



Fernanda Stamm Fronza

***MACHINE LEARNING* APLICADO NA USINAGEM PARA DETECÇÃO DE
DESGASTE EM FERRAMENTAS DE CORTE ATRAVÉS DA ANÁLISE DE
VIBRAÇÕES MECÂNICAS**

Horizontalina - RS

2022

Fernanda Stamm Fronza

***MACHINE LEARNING* APLICADO NA USINAGEM PARA DETECÇÃO DE
DESGASTE EM FERRAMENTAS DE CORTE ATRAVÉS DA ANÁLISE DE
VIBRAÇÕES MECÂNICAS**

Trabalho Final de Curso apresentado como requisito parcial para a obtenção do título de bacharel em engenharia de Controle e Automação na Faculdade Horizontina, sob a orientação do Prof. Me. Rodrigo Bastos.

Horizontina - RS

2022

FAHOR - FACULDADE HORIZONTINA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

A Comissão Examinadora, abaixo assinada, aprova o trabalho final de curso

**“*MACHINE LEARNING* APLICADO NA USINAGEM PARA DETECÇÃO DE
DESGASTE EM FERRAMENTAS DE CORTE ATRVÉS DA ANÁLISE DE
VIBRAÇÕES MECÂNICAS”**

**Elaborada por:
Fernanda Stamm Fronza**

Como requisito parcial para a obtenção do grau de Bacharel em
Engenharia de Controle e Automação

Aprovado em: 24/11/2022
Pela Comissão Examinadora

Prof. Me. Rodrigo Bastos
Presidente da Comissão Examinadora - Orientador

Prof. Me. Alexandre Kunkel da Costa
FAHOR – Faculdade Horizontina

Prof. Dr. Rafael Luciano Dalcin
FAHOR – Faculdade Horizontina

**Horizontina - RS
2022**

RESUMO

A indústria passou por muitas transformações ao longo do tempo, gerando inovação e impactando mudanças sociais e econômicas. Além disso, com a quarta revolução industrial, impulsionou-se o uso de tecnologias digitais com o intuito de agilizar a produção e a cadeia de suprimentos. Para garantir que os processos produtivos sejam ágeis e confiáveis, é necessário conhecê-los e monitorá-los para que os limites de funcionamento estejam dentro dos padrões estabelecidos. Na área da usinagem há muitas variáveis que devem ser consideradas para que o processo seja completo e sem danos. A utilização do *machine learning* conciliado com a manutenção preditiva permite prever falhas em equipamentos, fazendo com que seja possível planejar manutenções e a interferência no processo produtivo seja mínima. O presente projeto tem como objetivo realizar um estudo de caso em uma indústria do ramo de usinagem, a fim de detectar o desgaste em ferramentas de corte, durante o processo de usinagem, antes que ocorra a falha ou quebra da ferramenta. A partir dos dados de vibrações mecânicas, a análise foi feita utilizando a técnica de *machine learning*, tendo como resultado o monitoramento em tempo real do motor elétrico de um centro de torneamento.

Palavras-chave: *Machine Learning*. Usinagem. Manutenção preditiva.

LISTA DE FIGURAS

Figura 1 – Exemplos de métodos de medição indireta.....	13
Figura 2 – Fatores influentes sobre a qualidade superficial na usinagem de metais	14
Figura 3 – Classificação de vibração.....	16
Figura 4 – Evolução dos processos industriais	17
Figura 5 – Etapas do processo de <i>machine learning</i>	21
Figura 6 – Exemplo de <i>machine learning</i> supervisionado e não supervisionado	21
Figura 7 – Exemplo de <i>Machine learning</i> supervisionado	23
Figura 8 – Exemplo de <i>machine learning</i> com técnica de regressão	24
Figura 9 – Exemplo de <i>machine learning</i> com técnica de classificação	25
Figura 10 – Arduino UNO	26
Figura 11 – Microcontrolador ESP32	27
Figura 12 – ESP32 e suas conexões	28
Figura 13 – Plataforma IDE	29
Figura 14 – Acelerômetro 3 eixos MMA8452	30
Figura 15 – Fluxograma de etapas da aplicação do <i>machine learning</i>	32
Figura 16 – Esquema eletrônico.....	34
Figura 17 – Montagem do protótipo	35
Figura 18 – Conexão com a rede <i>wi-fi</i> pelo Arduino.....	36
Figura 19 – Leitura dos dados enviados pelo sensor	36
Figura 20 – Conversão de dados de 7 bits para 12 bits	37
Figura 21 – Definição das variáveis	38
Figura 22 – Cálculos de calibração do sensor.....	39
Figura 23 – Estrutura do <i>servidor Web</i>	40
Figura 24 – Exemplo da disposição dos dados após leitura no sensor.....	41
Figura 25 – Centro de Torneamento Ergomat TNG 32	42
Figura 26 – Dados técnicos do torno.....	43
Figura 27 – Fixação do sensor no motor.....	43
Figura 28 – Fluxo de trabalho no <i>orange data mining</i>	45
Figura 29 – Teste feito no modelo de regressão linear	45
Figura 30 – Gráfico gerado no <i>Orange</i> com os dados de aceleração	46
Figura 31 – Gráfico com acelerações acima do normal	47

LISTA DE QUADROS

Quadro 1 – Relação de escala aceleração	37
---	----

SUMÁRIO

1 INTRODUÇÃO	7
1.1 TEMA	8
1.2 DELIMITAÇÃO DO TEMA.....	8
1.3 PROBLEMA DE PESQUISA	8
1.4 HIPÓTESES.....	8
1.5 JUSTIFICATIVA	8
1.6 OBJETIVOS	9
1.6.1 Objetivo Geral	9
1.6.2 Objetivos Específicos	9
2 REVISÃO DA LITERATURA	10
2.1 USINAGEM INDUSTRIAL.....	10
2.2 MANUTENÇÃO PREDITIVA.....	11
2.3 MONITORAMENTO DE PROCESSOS DE USINAGEM.....	11
2.3.1 Variáveis de entrada e saída do processo	12
2.3.2 Monitoramento indireto	13
2.3.3 Análise de vibrações	15
2.4 INDÚSTRIA 4.0	17
2.4.1 Internet das coisas (IoT)	18
2.4.2 Cloud Computing	18
2.4.3 Big Data Analytics	19
2.4.4 Inteligência artificial	19
2.5 MACHINE LEARNING.....	20
2.5.1 Machine learning supervisionado	22
2.6 ARDUINO.....	25
2.6.1 ESP32	27
2.6.2 Software Arduino	28
2.7 ACELERÔMETRO	29
2.7.1 Acelerômetro 3 eixos MMA8452	30
2.8 NODE.JS.....	30
2.8.1 Node RED	31
2.9 ORANGE DATA MINING	31
3 METODOLOGIA	32
3.1 MÉTODOS E TÉCNICAS UTILIZADOS.....	32
3.2 MATERIAIS E EQUIPAMENTOS.....	33
4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	34
4.1 SIMULAÇÃO E MONTAGEM DO PROTÓTIPO	34
4.2 PROGRAMAÇÃO.....	35
4.2.1 Tratamento e processamento dos sinais de entrada	37
4.3 SERVIDOR WEB	39
4.4 INSTALAÇÃO DO PROTÓTIPO	41
4.5 APLICAÇÃO DO MACHINE LEARNING.....	44
CONCLUSÃO	48
REFERÊNCIAS	49
APÊNDICE	54

1 INTRODUÇÃO

A indústria passou por muitas transformações ao longo do tempo, gerando inovação e impactando mudanças sociais e econômicas. Além disso, com a quarta revolução industrial, impulsionou-se o uso de tecnologias digitais com o intuito de agilizar a produção e a cadeia de suprimentos. Dessa forma, Schwab (2019) afirma que a quarta revolução industrial transforma basicamente três pilares do mundo: o meio físico, o digital e o biológico, impactando na economia, negócios, sociedade, indivíduo e governo.

Neste processo de transformação digital das empresas, da indústria e dos modelos de negócio, a ciência de dados, a utilização de *big data*, juntamente com técnicas de estatística, inteligência artificial e *machine learning* são ferramentas fundamentais para os avanços tecnológicos no mundo digitalizado (GONÇALVES, 2020). O *machine learning*, como a própria tradução diz, tem como objetivo principal a aprendizagem, descobrir padrões ou fórmulas matemáticas que explicam a relação entre os dados coletados e analisados. Sendo assim, estuda formas de automatizar tarefas que seriam difíceis para o ser humano, utilizando algoritmos baseados em matemática e estatística (ESCOVEDO; KOSHIYAMA, 2020).

Diante disso, a utilização do *machine learning* visa aprimorar os sistemas produtivos, possibilitando um uso mais inteligente de produtos e serviços. Especificamente na indústria, o *machine learning* influencia na diminuição de custos de produção, eleva a qualidade dos produtos e serviços e diminui o desperdício (PREEZ; OOSTHUIZEN, 2019). O *machine learning* auxilia de forma automatizada na tomada de decisões, a partir da utilização de históricos de dados e integração com outras fontes.

A usinagem possui um conceito muito antigo, e é reconhecidamente o processo de fabricação mais popular do mundo. A operação de conferir a uma peça forma, dimensões e acabamento, através da remoção de material sob forma de cavaco, resume o processo de usinagem. Por ser um processo essencialmente prático, envolve um elevado número de variáveis. Segundo Schwab (2019), é praticamente impossível prever o desempenho de corte dos metais, porém isso não significa que o estudo detalhado dos processos de usinagem não tem o seu valor. Cada ponto fundamental que é detalhadamente estudado e interpretado de maneira adequada

contribui para o entendimento do processo, o procedimento mais próximo de prever seu comportamento.

Partindo do exposto, o presente estudo tem como objetivo detectar a falha ou desgaste em uma ferramenta, durante o processo de usinagem, antes que ocorra a falha ou quebra da ferramenta. Para isso, foi realizado o monitoramento das vibrações mecânicas do motor elétrico de indução de um centro de torneamento através da técnica de *machine learning*.

1.1 TEMA

Machine learning aplicado na usinagem para detecção de desgaste em ferramentas de corte a partir da análise de vibrações mecânicas.

1.2 DELIMITAÇÃO DO TEMA

Machine learning aplicado na usinagem para detecção de desgaste em ferramentas de corte nos centros de torneamento, a partir da análise de vibrações mecânicas de um motor elétrico, em uma indústria metalúrgica focada no ramo de prestação de serviços e produção de itens usinados, localizada no município de Horizontina, na região noroeste do estado do Rio Grande do Sul, no período de junho a novembro de 2022.

1.3 PROBLEMA DE PESQUISA

De que maneira a aplicação do *machine learning* no processo de usinagem auxilia na detecção do desgaste de ferramentas?

1.4 HIPÓTESES

- a) O *machine learning* contribui para a detecção do desgaste nas ferramentas durante o processo de usinagem através da análise de dados de vibrações mecânicas;
- b) As não conformidades em peças podem ser evitadas através do *machine learning*;
- c) Após a aplicação do *machine learning* haverá um aumento na produtividade.

1.5 JUSTIFICATIVA

Na indústria, os sistemas ciberfísicos e ferramentas como IOT, *machine learning*, *data analytics*, *big data*, inteligência artificial (IA), *cloud computing*, e

sistemas integrados, trazem oportunidades de melhoria na produtividade, redução de falhas e conseqüentemente redução de custos na manufatura (HORN, 2019), pois essas ferramentas auxiliam na tomada de decisão, a partir da análise de dados.

As empresas encaram um ambiente caracterizado por uma intensa competitividade a nível global e, por causa da exigência do mercado, fazem um esforço para atingir uma performance de alto nível. Parte desse esforço tem sido voltado para a manutenção, incluindo as funções de verificação dessas atividades. Neste contexto, a manutenção possui um papel essencial na garantia tanto da qualidade quanto da produtividade empresarial (MARCORIN; LIMA, 2003).

Visando melhorar a detecção e reconhecimento de padrões de falha, propõem a utilização de ferramentas de inteligência artificial. Esta área de monitoramento do estado das máquinas é foco de um interesse crescente, tanto da comunidade científica quanto das indústrias, assim como a utilização de inteligência artificial como ferramenta de previsão de falhas (VELLOSO, 2020).

Frente ao exposto, buscando justificar a importância da manutenção preditiva, este projeto visa implementar o *machine learning* em processos de usinagem para detectar desgaste em ferramentas, a fim de aumentar a produtividade, evitar não conformidades, desperdícios, e diminuir custos de produção.

1.6 OBJETIVOS

1.6.1 Objetivo Geral

Demonstrar como o *machine learning* contribui para a detecção do desgaste em ferramentas no processo de usinagem.

1.6.2 Objetivos Específicos

- Monitorar as vibrações mecânicas dos centros de torneamentos durante a usinagem;
- Determinar o ponto de troca das ferramentas durante a usinagem antes da quebra aplicando o *machine learning*;
- Aumentar a produtividade na usinagem a partir da diminuição do *setup* ou parada de máquina.

2 REVISÃO DA LITERATURA

2.1 USINAGEM INDUSTRIAL

A usinagem é reconhecidamente o processo de fabricação mais popular do mundo, transformando em cavacos em torno de 10% de toda a produção de metais e empregando dezenas de milhões de pessoas (MACHADO *et al.*, 2015). Apesar disso, é um processo bastante imprevisível, pois envolve um processo complexo, onde é difícil determinar as condições ideais de corte e simples ao mesmo tempo, pois uma vez determinadas estas condições, o cavaco se forma corretamente. As condições ideais de corte, segundo Machado (2015), são aquelas capazes de produzir peças dentro de especificações de forma, tamanho e acabamento ao menor custo possível.

De acordo com a ABNT NBR 6175 (1971), a usinagem é um processo mecânico mediante a remoção de cavaco por determinada ferramenta, a todos que visa conferir forma a uma peça, as suas dimensões e acabamentos especificados, ou uma combinação qualquer destes três itens. Segundo Ferraresi (1970), o cavaco pode ser definido como uma porção de material removido da peça por uma ferramenta, o qual apresenta forma geométrica irregular. Também pode-se citar a definição pela norma DIN 8580, a qual diz que a usinagem se aplica a todos os processos em que há remoção de materiais, sendo estes definidos em forma de cavaco (DIN 8580, 2009, apud, PERES *et al.*, 2017).

Segundo Sacomano *et al.* (2018), o processo de usinagem pode ser classificado em dois grupos, considerando a geometria da ferramenta empregada:

- Usinagem com ferramenta de geometria definida: utilizada para fresar, torner, alargar, serrar, furar, rosquear, brochar e plainar;
- Usinagem com ferramenta de geometria não definida: utilizada para lixar, retificar, jatear, tamborear, polir, lapidar e brunir.

Além de ser importante para a indústria, a usinagem representa uma grande parte do processo de fabricação de itens importantes, como por exemplo a fabricação de um trator, quando detalhado todo o projeto, pode-se analisar que mais da metade dos componentes utilizados passam pelo processo de usinagem (SANTOS; SALES, 2007).

Assim como em outros segmentos, o setor de usinagem precisa acompanhar a evolução das tecnologias, buscando alto desempenho e qualidade em seus produtos.

A evolução que a indústria 4.0 trouxe para o mundo, permitiu que se torne possível determinar novas soluções e estratégias de negócios (ALVES, 2019).

2.2 MANUTENÇÃO PREDITIVA

Segundo Francisco (2019), a manutenção tradicionalmente pode ser dividida de acordo com a forma de programação e o objetivo das tarefas a serem executadas. Com relação à programação, são comuns as classes de manutenção programada e não programada, respectivamente, as atividades são designadas obedecendo critérios de tempo e condições pré-definidas e as executadas em função da necessidade. Quanto ao objetivo, geralmente são definidos quatro tipos de manutenção: corretiva, preventiva, preditiva e detectiva (PINTO; XAVIER, 2010)

“Preditiva: é a atuação realizada com base na modificação de parâmetros de condição ou desempenho, cujo acompanhamento obedece a uma sistemática. A manutenção preditiva também é conhecida como manutenção sob condição ou manutenção com base no estado do equipamento. É baseada na tentativa de definir o estado futuro de um equipamento ou sistema, por meio dos dados coletados ao longo do tempo por uma instrumentação específica, verificando e analisando a tendência de variáveis do equipamento (PINTO; XAVIER, 2010).”

Segundo Borlido (2017), a manutenção preditiva é a manutenção centrada num conjunto de ações que levam à redução de falhas de funcionamento. O seu principal enfoque é o de evitar a manutenção corretiva, pois através de um correto monitoramento das condições das máquinas e parâmetros, é possível prever prováveis falhas. Seu principal objetivo é determinar a manutenção a realizar em determinada peça, de determinada máquina. A prática de uma manutenção preditiva, ajuda na redução de custos em manutenções preventivas e corretivas.

2.3 MONITORAMENTO DE PROCESSOS DE USINAGEM

Nas últimas décadas, os processos de usinagem passaram por profundas mudanças visando principalmente o aumento da produtividade e a garantia da qualidade na indústria metal mecânica (SOUZA, 2015). Ainda, segundo Souza (2015), esta busca prioriza a redução de tempos de produção e a melhoria do acabamento usinado, proporcionando uma evolução natural de máquinas e equipamentos. A utilização de sistemas de monitoramento e controle que interagem com a máquina-ferramenta com comando numérico é uma tendência crescente dessa evolução, o que

acarreta a automatização do processo e redução dos tempos improdutivos e os custos agregados.

As interrupções nos processos são necessárias para correções decorrentes de uma imprevista substituição do inserto por motivo de quebra ou por uma troca prematura mesmo. Essa manutenção corretiva é geralmente realizada a partir de uma detecção de falhas no processo pelo operador e através da observação de peças acabadas (SOUZA, 2015).

Com isso, surgiu a necessidade de desenvolver sistemas que fossem capazes de supervisionar o processo de usinagem através do monitoramento em tempo real das grandezas físicas envolvidas, permitindo a diminuição da intervenção e substituição de algumas funções do operador (SOUZA, 2015).

2.3.1 Variáveis de entrada e saída do processo

No contexto geral, as empresas baseiam-se em dados estatísticos para troca de ferramentas. Mas, devido ao grau de complexidade e características imprevisíveis na usinagem fazem com que a variação do tempo de vida da ferramenta seja muito grande (SOUZA, 2015). Portanto, muitas ferramentas são trocadas em condições de serem mantidas em operação.

Atualmente, o monitoramento de ferramentas é uma condição indispensável para a melhoria no desempenho de processos de fabricação de componentes metálicos. Ele garante a detecção de falhas de funcionamento da ferramenta utilizada na máquina, como a ferramenta lascada ou com desgaste excessivo, e a prevenção de falhas para proteger a ferramentas, a máquina e a peça contra colisões e danos no processo, resultantes da falha detectada (SOUZA, 2015).

Uma maneira de controlar o processo de usinagem é atuar diretamente nas variáveis de entrada (independentes), como características da peça (material, geometria, dureza, etc.), características da ferramenta de corte (geometria e material), parâmetros de corte (avanço por volta, avanço por dente, profundidade do corte axial, profundidade do corte radial, rotação, velocidade de corte, meio lubrificante e máquina-ferramenta) e mensurar seu efeito através da medida de variáveis de saída (dependentes) mais importantes, como tipo e forma do cavaco; força e potência de usinagem; vibração; temperatura da região de corte; falhas na ferramenta de corte; acabamento da superfície usinada.(SOUZA, 2015)

“A otimização dos processos de usinagem, nos diversos setores desta, sempre foi algo essencial para garantir o aumento da produtividade e a qualidade dos produtos fabricados em larga escala. Desde um melhor aproveitamento da ferramenta (visando aumentar sua vida útil) até a utilização de parâmetros de entrada adequados para cada tipo de usinagem, têm-se diferentes objetos de estudo no campo acadêmico. Procura-se, portanto, melhorar os processos de usinagem sem aumentar os tempos de parada para troca de ferramentas, o que acarretaria um aumento no custo do produto (SOUZA, 2015).”

Segundo Souza (2015), um sistema confiável de monitoramento permite, por exemplo, estimar as condições de rugosidade da superfície usinada ou o estado da ferramenta. Além disso, possibilita otimizar os parâmetros de corte, minimizar os estoques, diminuir os tempos de parada de máquinas, comparar o desempenho de ferramentas e procedimentos de fabricação.

2.3.2 Monitoramento indireto

O monitoramento indireto baseia-se no processo de controle em tempo real, ou seja, simultaneamente ao processo de medição das falhas da ferramenta. Este método consiste em acompanhar e avaliar uma grandeza física, mensurável, que pode ser correlacionada com o desgaste da ferramenta durante o processo (SOUZA, 2015). A Figura 1 abaixo apresenta os principais métodos de medição das condições da ferramenta em usinagem.

Figura 1 – Exemplos de métodos de medição indireta

Método	Procedimento	Medição	Transdutor/Sensor
INDIRETO	Força de usinagem	Alterações no comportamento da força.	Dinamômetros: piezelétricos ou resistivos (<i>strain-gage</i>); sensor de torque no eixo-árvore.
	Emissão acústica ⁴	Ondas de deformação plástica.	Transdutor piezelétrico.
	Som	Ondas acústicas.	Microfone.
	Vibração	Oscilações da ferramenta e/ou suporte.	Acelerômetro piezelétrico.
	Temperatura	Variação da temperatura na região de corte.	Termopar; pirômetro; reflexão da superfície do cavaco ou sua coloração.
	Potência	Potência ou corrente consumida no motor (avanço ou rotação).	Amperímetro; sensor de corrente; sensor de potência.
	Acabamento da superfície usinada	Mudanças na rugosidade da superfície.	Rugosímetro; laser; métodos ópticos: sensor de fibra óptica, câmera CCD.

Fonte: Adaptação Souza, 2015

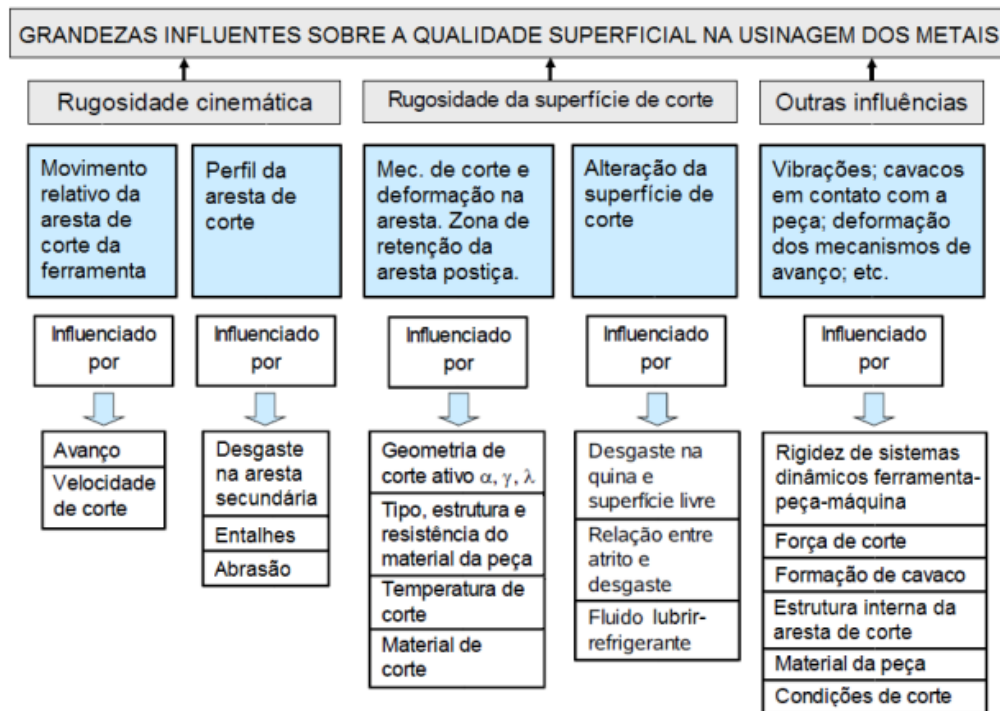
Os métodos indiretos de monitoramento baseiam-se no fato de que o estado da ferramenta de corte está associado a muitos fenômenos físicos relacionados com

o material usinado, o tipo de ferramenta, as condições do processo etc. Estes fenômenos podem ser detectados através de transdutores que, posicionados em locais estratégicos da máquina, qualificam indiretamente as falhas, principalmente por sinais de força, potência, emissão acústica, ultrassom, vibração ou temperatura gerados em processos de fabricação por usinagem (SOUZA, 2015).

2.3.2.1 Falhas na ferramenta de corte

No processo de usinagem, em função das condições de corte, a ferramenta fica sujeita a uma combinação de vários fatores de origem térmica, mecânica, química e abrasiva (SOUZA, 2015). Como nem o material das peças nem o da ferramenta de corte é homogêneo, uma série de avarias e desgastes de naturezas distintas podem ser observadas na ferramenta durante a utilização, conforme ilustra a Figura 2.

Figura 2 – Fatores influentes sobre a qualidade superficial na usinagem de metais



Fonte: Souza, 2015, p. 8

Segundo Souza (2015), ferramentas desgastadas geram qualidades insatisfatórias na superfície usinada, assim como erros dimensionais de forma e tamanho da peça. Além disso, o risco de quebra da ferramenta aumenta. Sendo assim, a detecção automática de ferramentas desgastadas é essencial para garantir um processo livre de erros. Devido ao fato de a qualidade da peça ser o ponto mais

importante, não basta coletar informações somente do estado da ferramenta, mas também o acabamento gerado na peça.

2.3.3 Análise de vibrações

As vibrações de natureza mecânica são fenômenos importante do mundo físico, e suas manifestações ocorrem com frequência no universo que nos circunda, liberando muitas vezes grandes quantidades de energia, como ocorre nos tremores de terra na crosta terrestre (JUNIOR, 2006).

A vibração mecânica se manifesta na presença de esforços de natureza dinâmica, isto é, em que a intensidade ou direção das forças aplicadas nos componentes mecânicos muda continuamente com o tempo. As forças centrífugas são exemplos que atuam sobre um eixo (rotor) quando existe algum desbalanceamento de massas devido a assimetrias geométricas (JUNIOR, 2006).

Na indústria, as medições de vibrações são necessárias por diversos motivos. A grande demanda de produtividade, a otimizações de processos, a competitividade e a elevada busca por redução de custos nos processos culminam em maior velocidade de operação por máquina envolvidas no processo e necessidade de materiais mais leves (NETO *et al.*, 2021).

É fundamental a medição periódica da vibração das máquinas e estruturas, a fim de garantir que os equipamentos sempre estejam trabalhando dentro da margem de segurança. Qualquer anomalia ou alteração nos valores de frequências naturais deve ser analisada com cuidado e atenção, pois isso pode indicar uma falha ou necessidade de manutenção (NETO *et al.*, 2021).

“Os fenômenos vibratórios de uma máquina são resultado das forças dinâmicas que a excitam. Essas vibrações se propagam por todos os sistemas das máquinas, bem como para as estruturas interligadas a ela. Geralmente uma máquina vibra em várias frequências e amplitudes correspondentes. Os efeitos de uma vibração severa são o desgaste e a fadiga, certamente responsáveis por quebras de máquinas e equipamentos (NETO *et al.*, 2021).

Os motores elétricos, que possuem a velocidade de rotação padrão conforme o número de polos, podem variar de 900 RPM à 3600 RPM. Os espectros são analisados no domínio da frequência, em ciclos por segundo, é necessário realizar a conversão de RPM para Hertz. Isso é feito a partir da divisão da rotação em RPM por 60 Hz (NETO *et al.*, 2021). Sendo assim, tem-se:

- 900 rpm = 15 Hz
- 1200 rpm = 20 Hz
- 1800 rpm = 30 Hz
- 3600 rpm = 60 Hz

O estado operacional das máquinas elétricas rotativas é classificado de acordo com a ISO 10816 (2016), a partir dos níveis de vibração. Os quatro intervalos de estado da máquina são classificados em A, B, C e D, que são:

- Nível A: equipamento novo ou em perfeitas condições de operação;
- Nível B: equipamento em boas condições de operação;
- Nível C: equipamento sob condições operacionais permitidas, mas em níveis de manutenção e alerta;
- Nível D: equipamento em condições de operação não permissíveis com níveis de vibração ultrapassam o limite permitido.

Conforme a norma, os níveis de severidade são apresentados de acordo com a Figura 3.

Figura 3 – Classificação de vibração

R.m.s Velocidade de Vibração mm/s	CLASSE I	CLASSE II	CLASSE III	CLASSE IV
0,28	A	A	A	A
0,45	A	A	A	A
0,71	A	A	A	A
1,12	B	A	A	A
1,8	B	B	A	A
2,8	C	B	B	A
4,5	C	C	B	B
7,1	D	C	C	B
11,2	D	D	C	C
18	D	D	D	C
26	D	D	D	D
45	D	D	D	D

Os motores e máquinas elétricas são classificados de acordo com o com sua potência e tamanho:

- Classe I: motores individuais acoplados a máquinas individuais operando em condições normais, de até 15 kW;
- Classe II: máquinas de tamanho médio porte, de 15 a 75 kW, sem fundações especiais ou máquinas de até 300 kW em fundações especiais;
- Classe III: movimentadores primários grandes e outras máquinas com grandes massas rotativas;
- Classe IV: movimentadores primários grandes e outras máquinas rotativas grandes, como turbo geradores e turbinas a gás maiores que 10 MW.

2.4 INDÚSTRIA 4.0

Desde o final do séc. XVIII, a indústria tem passado por transformações que revolucionaram a maneira como os produtos são fabricados e trouxeram vários benefícios, especialmente no que tange o aumento da produtividade. A chamada manufatura inteligente das indústrias foi nomeada como a quarta revolução industrial ou a Indústria 4.0, o crescimento e amadurecimento evolutivo das novas Tecnologias de Informação e Comunicação (TIC), aliada a processos e produtos industriais impulsionaram uma modificação de modelo industrial (LINDSTROM; *et al.*, 2019). Na Figura 4, pode-se verificar a evolução dos processos industriais ao longo do tempo.

Figura 4 – Evolução dos processos industriais



Fonte: Silveira, 2016

Segundo Coelho (2016), o impacto da Indústria 4.0 vai além da simples digitalização, passando por uma forma muito mais complexa de inovação baseada na

combinação de múltiplas tecnologias, que forçará as empresas a repensar a forma como gerem os seus negócios e processos, como se posicionam na cadeia de valor, como pensam no desenvolvimento de novos produtos e os introduzem no mercado, ajustando as ações de *marketing* e de distribuição.

Para a ciência de dados, essa transformação na indústria possibilita a retirada de conhecimento importante de trabalhos monitorados, admissão de procedimentos eficientes de monitoramento e agregação de dados, assim como a utilização de métodos de *machine learning* e aperfeiçoamento. Um dos fundamentos da ciência de dados nesta situação é antecipar com melhor precisão os procedimentos no maquinário industrial, ferramentas e processos, com isso conseguindo prever eventualidades e danos, que porventura podem causar graves perdas em questões econômicas e de segurança. Nessa situação, a estimativa orientada por dados está gradativamente conquistando atenção em vários setores das indústrias (DIEZ-OLIVAN *et al.*, 2019).

2.4.1 Internet das coisas (IoT)

O conceito de internet das coisas está relacionado com o fato de que qualquer equipamento conectado à internet pode estar conseqüentemente conectado a outro (BORLIDO, 2017). Uma das vantagens da internet das coisas é desenvolver uma comunicação entre os aparelhos ou as máquinas incrementando a eficiência, a segurança industrial e diminuindo custos e tempos. Entre os exemplos mais conhecidos podem-se destacar os *smartphones* conectando carros inteligentes e automação residencial, por exemplo.

O IoT deve permitir uma conexão autônoma e segura para troca de dados entre dispositivos e aplicações do mundo real, utilizando a internet para comunicação, troca de informações, tomada de decisões e invocação de ações. Esse tema é muito relevante em instituições de ensino, na indústria e nas organizações governamentais, pois o IoT, possui potencial significativo para melhorar questões sociais, profissionais e econômicas (MORAIS *et al.*, 2018).

2.4.2 Cloud Computing

A computação na nuvem é assim chamada porque não é possível saber onde os servidores que processam e armazenam os dados estão localizados, sendo assim

ela é fundamental para que essas informações possam ser acessadas, de forma fácil, de qualquer parte do mundo que haja internet (SACOMANO *et al.*, 2018).

Cloud Computing, ou computação na nuvem, consiste em acessar aplicações ou arquivos por meio de páginas da internet ou programas específicos de conexão remota. Grandes centros computacionais, como centro de processamento de dados (data centers), estão espalhados pelo mundo, para favorecer os diversos tipos de conexão existentes. Além disso, é possível contar com o compartilhamento entre máquinas de usuários, que funcionam como uma nuvem (SILVA *et al.*, 2020).

2.4.3 Big Data Analytics

Chama-se *big data* um conjunto de informações geradas por um sistema, seja ele produtivo, comercial, marketing e outros, que precisa ser minuciosamente analisada, pois há detalhes que podem significar o sucesso de qualquer organização, desde que sejam bem utilizados (SACOMANO *et al.*, 2018).

“Os dados estruturados, ou seja, os que são passíveis de serem enquadrados dentro de uma estrutura organizacional, já contam com análises feitas por vários métodos estatísticos. A complexidade de análise da big data cresce à medida que incorpora a análise de dados não estruturados, como imagens, expressões faciais, sons, documentos digitalizados etc. (SACOMANO *et al.*, 2018, p.38).”

Big data analytics serve de forma dinâmica para tomada de decisões. Seu principal objetivo é o de ajudar as empresas a tomar melhores decisões na hora adequada (BORLIDO, 2017).

2.4.4 Inteligência artificial

A inteligência artificial pode ser definida como a capacidade de aprendizagem das máquinas previamente programadas, utilizando de algoritmos bem elaborados e complexos que proporcionem a tomada de decisões, especulações e até intenções baseadas nos dados fornecidos. No entanto, a IA pode ser dividida em camadas ou em partes que a compõem, e dessa forma é introduzido os conceitos de *machine learning* e *deep learning* (DAMACENO; VASCONCELOS, 2018).

A aplicação da inteligência artificial é ampla em diversas áreas, como por exemplo, na educação, onde os avanços no processamento de linguagem natural, combinados com o aprendizado de máquina, permitiram a classificação automática de tarefas, bem como a compreensão orientada por dados das necessidades individuais de aprendizado dos alunos. Além disso, existem aplicações em finanças,

busca de empresa, indústria, hospitais e medicina, mídia e comércio eletrônico, brinquedos e jogos, transporte, etc. (MORAIS *et al*, 2018).

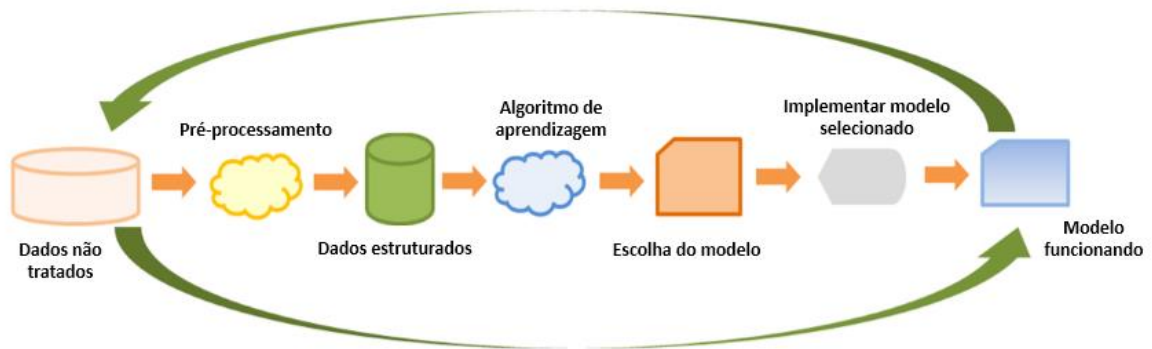
2.5 MACHINE LEARNING

A indústria 4.0 vem permitindo que empresas de manufatura atinjam seus objetivos com muito sucesso. Num ambiente onde inúmeros sensores estão conectados e enviando informações em tempo real, a Internet das Coisas Industrial tem contribuído para que novas oportunidades sejam incorporadas no chão de fábrica (ALVES, 2019). Ainda segundo Alves (2019), a inteligência artificial (IA) e o *machine learning* são uma das principais tecnologias que estão embarcadas nessa onda dentro da indústria. O acesso aos dados da produção tornou-se um recurso extremamente importante, pois através do uso de algoritmos de IA, especialmente *machine learning*, os fabricantes podem melhorar a eficiência da linha de produção, qualidade do produto e segurança dos funcionários.

A manutenção é um dos grandes vilões sobre os custos de operação. Um recurso comum entre os fabricantes para reduzir esse tipo de gasto é prever o comportamento de uma máquina. Algumas empresas implementam a manutenção preventiva através do sistema supervisório SCADA, onde configurações e regras estáticas de alerta são feitas pelo homem. Na grande maioria das vezes, esse processo manual não considera comportamentos complexos das máquinas ou do processo como um todo. Diante disso, alertas são disparados erroneamente com falsos-positivos ou não acusam falhas reais. Dessa forma, é exatamente nesse ponto onde as técnicas de *machine learning* são aplicadas (ALVES, 2019).

Machine learning, como o próprio nome já sugere, é o processo de aprendizagem contínuo de máquina. Consiste basicamente em fornecer dados de entrada e assim a máquina pode aprender com esses dados e elaborar saídas que satisfaçam a situação problema (DAMACENO; VASCONCELOS, 2018). Os algoritmos utilizados no *machine learning* são estruturados com equações pré-definidas para organizar e executar dados conforme a demanda. Na Figura 5, pode-se verificar as etapas de um processo de *machine learning*.

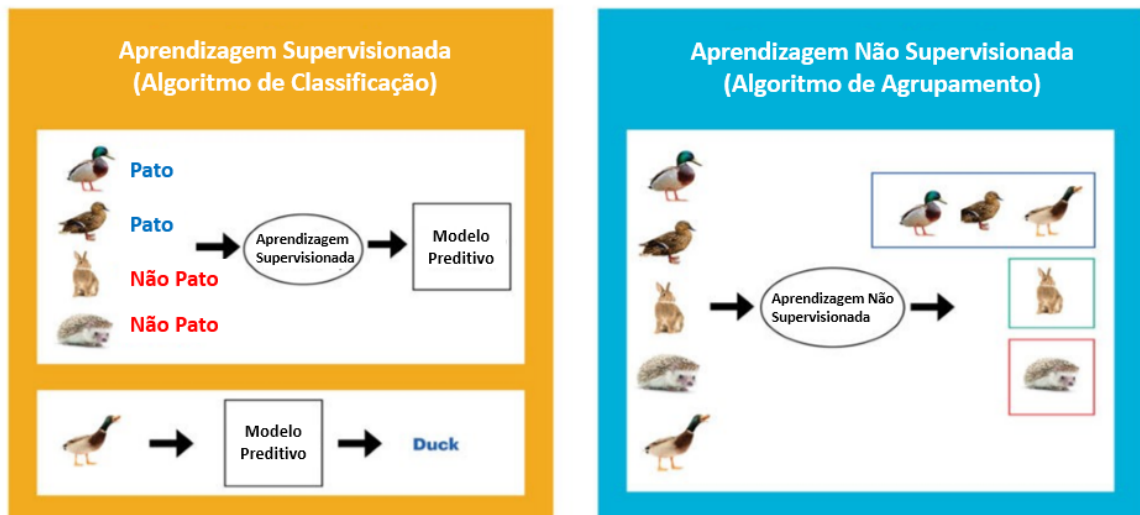
Figura 5 – Etapas do processo de *machine learning*



Fonte: Adaptado Damaceno, 2018

Os algoritmos de *machine learning* utilizam dados do chão de fábrica como sensores supervisórios, scanners e dados de produção fornecidos pelo sistema ERP, MES e qualidade. O processo de treinamento de máquina permite que o algoritmo detecte anomalias e forneça correlação entre dados, buscando e sugerindo padrões no processo fabril (ALVES, 2019). Há dois tipos de técnicas utilizadas em *machine learning*: supervisionada e não supervisionada, conforme exemplificado na Figura 6.

Figura 6 – Exemplo de *machine learning* supervisionado e não supervisionado



Fonte: Adaptado Kenji, 2019

Conforme a Figura 6, pode-se verificar que a aprendizagem supervisionada utiliza métodos de entrada com dados rotulados e usam esses dados e seus atributos para determinar um novo conjunto de dados desconhecidos. Já a aprendizagem não

supervisionada, agrupa objetos de acordo com as suas características (MORAIS *et al*, 2018).

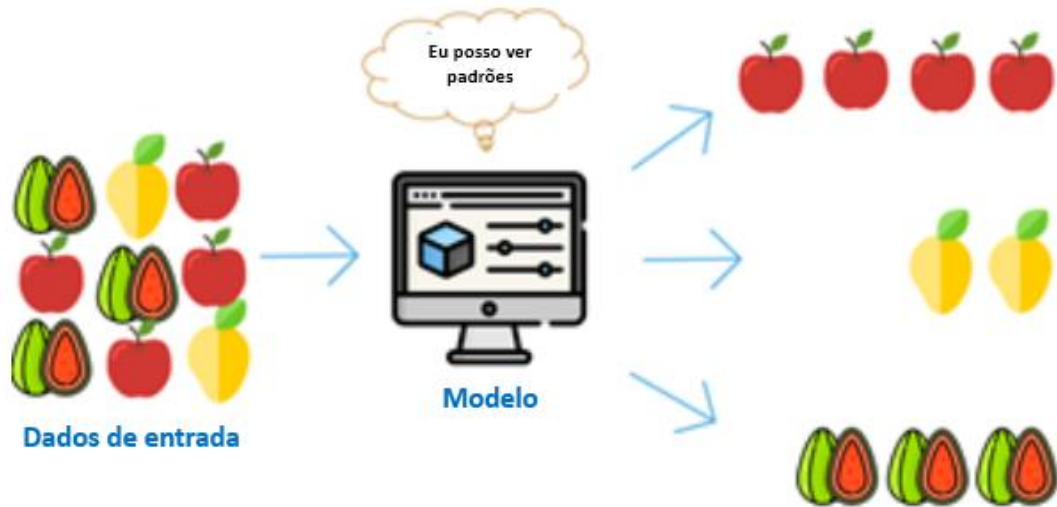
Um dos principais fatores para o uso de técnicas de *machine learning* é a capacidade de analisar elevada quantidade de dados em tempo real e fornecer as respostas rapidamente. A produção é constantemente analisada e a previsão de falhas permite reduzir tempo de parada e desperdícios, além de melhorar a qualidade do produto e as taxas de falhas (ALVES, 2019).

2.5.1 *Machine learning* supervisionado

Em aprendizado supervisionado, algoritmos são utilizados para induzir modelos preditivos por meio da observação de um conjunto de objetos rotulados (VON LUXBURG; SCHÖLKOPF, 2008), tipicamente referenciado como conjunto de treinamento.

A técnica mais comum do *machine learning* supervisionado consiste no princípio básico onde têm-se os dados de entrada e de saída e deseja-se mapear a função que conecta as duas variáveis. Uma de suas principais características é a interação com o homem, pois exige treinamento, definição de algoritmos, ajuste de dados, criação de mapas de visualização etc. A partir destes dados é possível mapear a equação que permitirá prever a saída, a partir de novas variáveis de entrada do sistema. O principal caso na manufatura envolvendo *machine learning* é a manutenção preditiva e pode ser feito usando-se dois tipos de técnicas supervisionadas: regressão e classificação (ALVES, 2019). Na Figura 7 tem-se um exemplo de *machine learning* supervisionado, onde os dados são rotulados e após isso são classificados de acordo com os seus atributos.

Figura 7 – Exemplo de *Machine learning* supervisionado



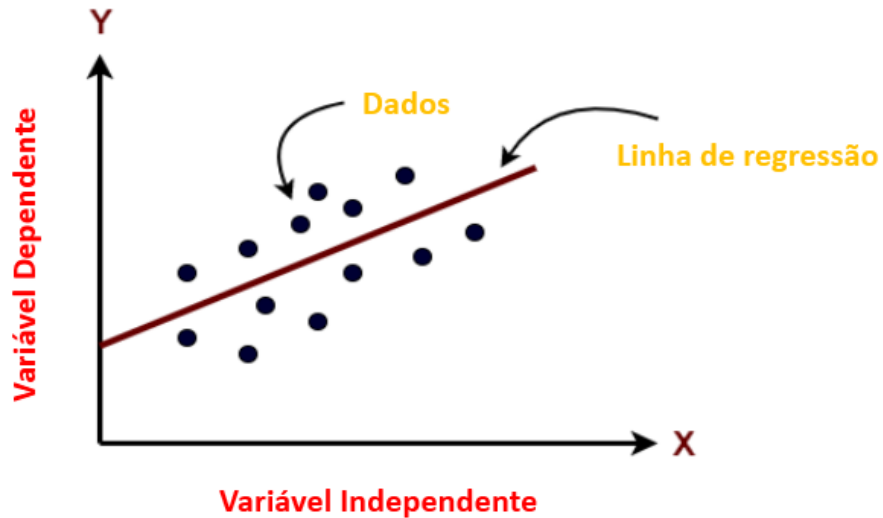
Fonte: Heidenreich, 2018

A característica básica de sistemas de aprendizado supervisionado é que os dados que utilizamos para treiná-los contém a resposta desejada, isto é, contém a variável dependente resultante das variáveis independentes observadas (HONDA; *et al.*, 2017). A área do *machine learning* supervisionado é a que concentra a maioria das aplicações bem-sucedidas e onde a maioria dos problemas já estão bem definidos.

2.5.1.1 Regressão

Quando um valor é contínuo, ou seja, algo esperado, o aprendizado supervisionado é chamado de regressão. Por exemplo, como acontece no preço de cotações, baseado em um conjunto de dados, o algoritmo pode prever o índice da inflação para um determinado período (VIEIRA, 2018). Na Figura 8 tem-se um exemplo de regressão linear.

Figura 8 – Exemplo de *machine learning* com técnica de regressão



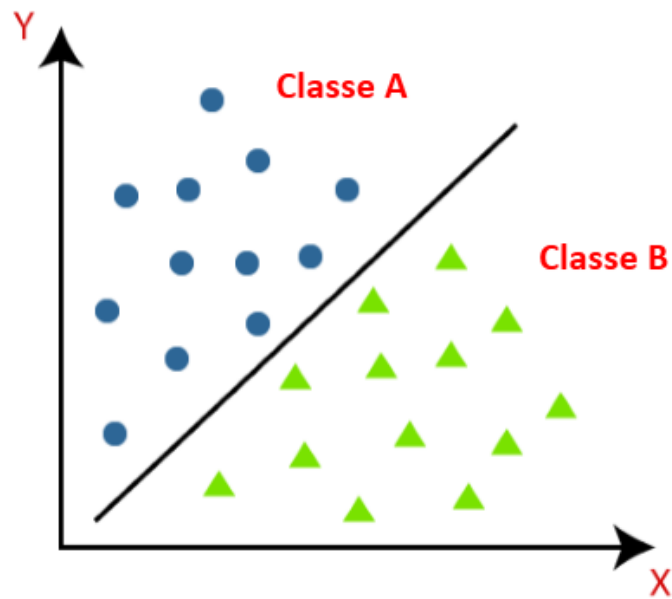
Fonte: Alves, 2019

Conforme a Figura 8, pode-se verificar que a técnica supervisionada de regressão é utilizada quando os dados possuem grandezas conhecidas e variam dentro de uma faixa específica, como por exemplo temperatura, peso, velocidade etc. Na manufatura, a técnica de regressão é usada para estimar a vida útil da máquina, fazendo previsão de quantos ciclos restam até a máquina ou ferramenta falhar (ALVES, 2019).

2.5.1.2 Classificação

Em classificação o algoritmo vai criar um classificador capaz de aprender com informações já apresentadas no treinamento, com a finalidade de dizer se um objeto pertence ou não a uma determinada classe. Os e-mails utilizam um exemplo de classificação para determinar se um e-mail é ou não “spam” (VIEIRA, 2018). Na Figura 9 pode-se observar um exemplo de *machine learning* com técnica de classificação.

Figura 9 – Exemplo de *machine learning* com técnica de classificação



Fonte: Alves, 2019

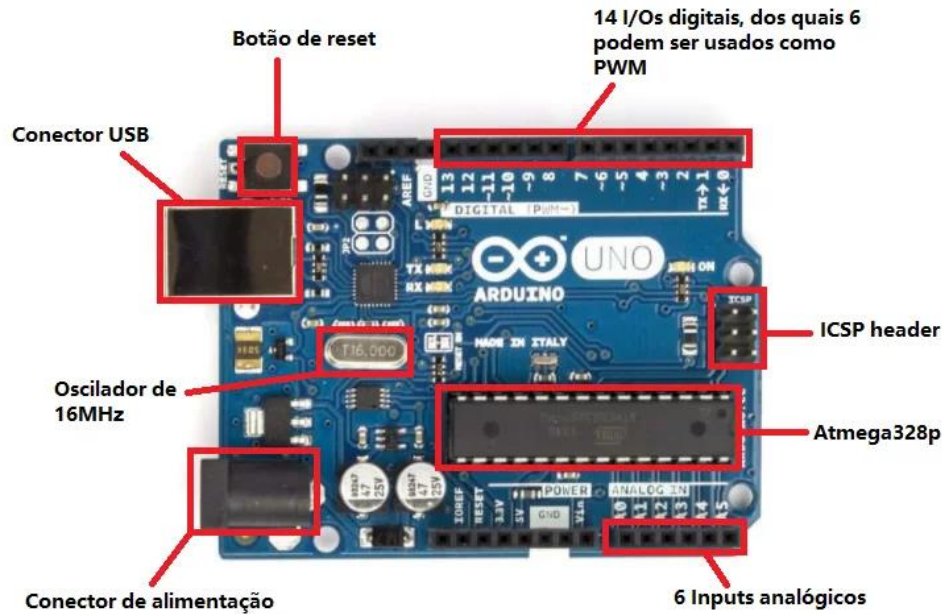
Na técnica de classificação os dados são associados um objeto a uma classe específica, onde é calculado a probabilidade de algo após as variáveis terem sido caracterizadas anteriormente. Segundo Velasquez, a classificação baseia-se em prever a categoria de uma dada observação. Para a classificação, a acurácia e o erro são as medidas básicas de desempenho, quanto maior a acurácia do modelo, maior serão os acertos e menor serão os erros cometidos. São exemplos de técnicas de classificação, a árvore de decisões, o *k-Nearest Neighbors* e a curva ROC.

2.6 ARDUINO

Arduino é uma plataforma de prototipagem eletrônica *open source* que utiliza microcontrolador Atmel AVR com circuitos de entrada/saída que podem ser conectados a um computador e programada via IDE e utiliza linguagem baseada em C/C++ (ARDUINO, 2018).

Segundo Lousada (2022), a composição do *hardware* Arduino pode variar de acordo com a versão e modelo da plataforma eletrônica, mas em geral as placas possuem a seguinte estrutura física, conforme a Figura 10.

Figura 10 – Arduino UNO



Fonte: Lousada, 2022

Os principais motivos para a utilização de uma plataforma Arduino em projetos são o baixo custo de prototipagem, a disponibilidade de *softwares* de simulações gratuitas, a fácil programação, o grande número de tutoriais, artigos e projetos prontos disponíveis na internet, a extensa comunidade de desenvolvedores além disso, não requer experiência ou grandes conhecimentos prévios de eletrônica e programação etc. (LOUSADA, 2022).

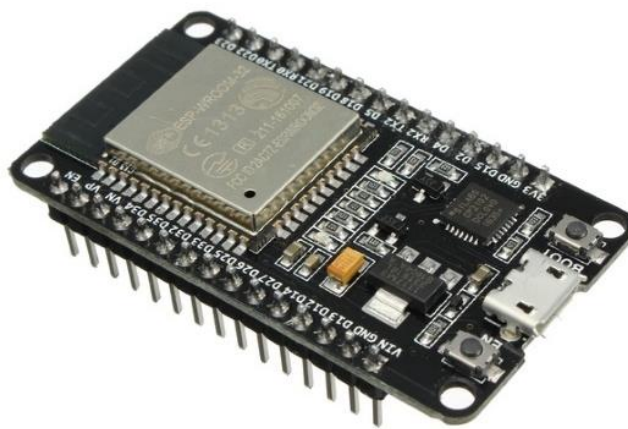
Existem diversos modelos de plataformas de prototipagem Arduino disponíveis no mercado atualmente, e a escolha do usuário vai depender da complexidade e da funcionalidade do mesmo (LOUSADA, 2022). Entre as classes disponíveis, cita-se as seguintes:

- *Entry Level*: versões mais simples e focadas em quem está começando a desenvolver projetos eletrônicos;
- *Enhanced Features*: placas com funcionalidades voltadas para projetos mais complexos;
- *Internet of Things*: plataformas específicas para projetos de internet das coisas;
- *Education*: voltado exclusivamente para o ensino de eletrônica e programação;
- *Wearable*: plataformas com características específicas para o desenvolvimento de dispositivos *wearables*, como por exemplo, *smartwatches*;
- *3D printing*: plataformas para impressoras 3D desenvolvidas com Arduino.

2.6.1 ESP32

O microcontrolador ESP32 foi desenvolvido em 2016 pela empresa desenvolvedora de tecnologia *Espressif Systems*, sendo que ele vem sendo considerado um dos mais robustos e notáveis controladores do mercado. Sua tecnologia tem fortes características na velocidade de processamento, acessibilidade e conectividade, principalmente pela sua inteligência com conexão *wi-fi* (KOLBAN, 2018). Na Figura 11, pode-se verificar a imagem de um microcontrolador ESP32.

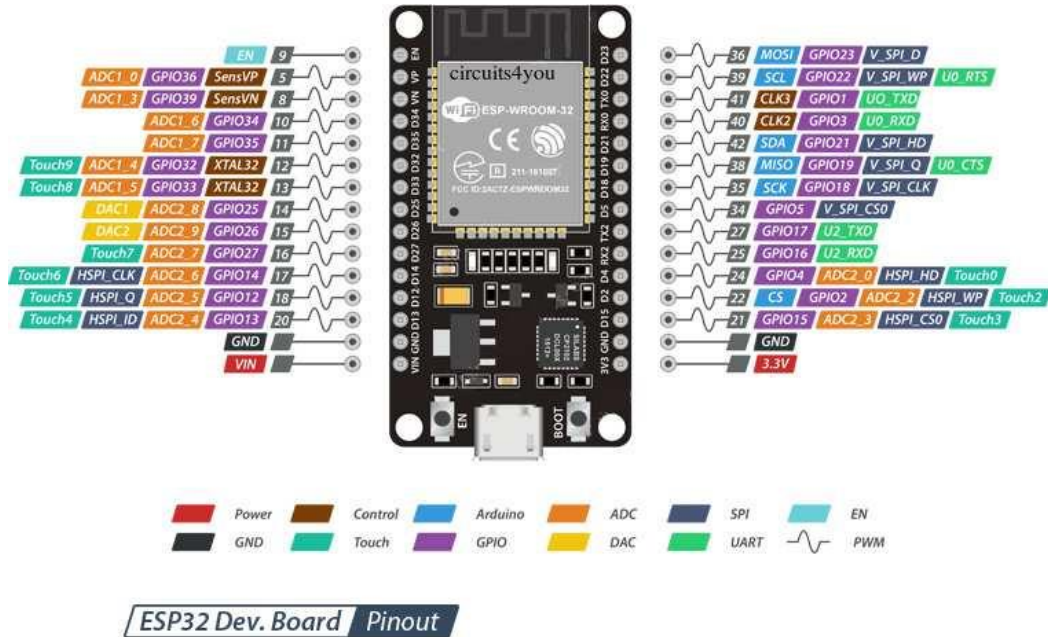
Figura 11 – Microcontrolador ESP32



Fonte: FilipeFlop, 2022

O ESP32 possui um processador de modelo single ou dual-core de 32 bits (com dois núcleos físicos de processamento), podendo chegar a trabalhar com 240 MHz de frequências de *clock*. Comparado com outros processadores, o ESP32 possui uma vantagem enorme com relação a sua capacidade de armazenamento (IBRAHIM, 2017). Na Figura 12 estão representados seus componentes e as distribuições dos pinos.

Figura 12 – ESP32 e suas conexões



ESP32 Dev. Board Pinout

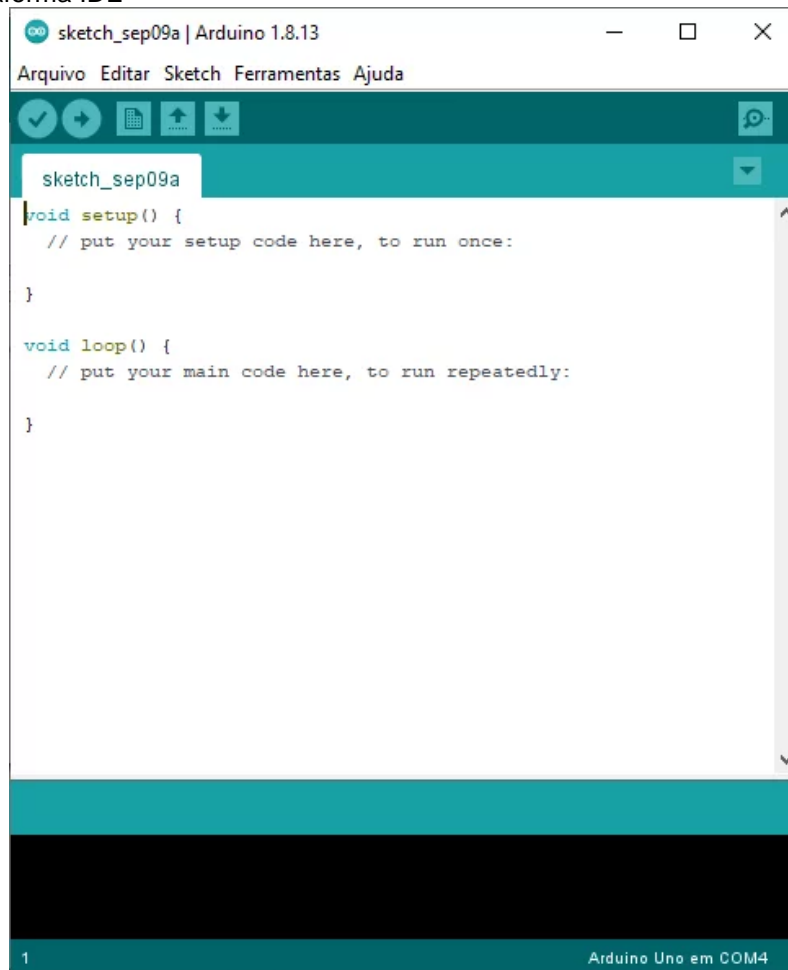
Fonte: Circuits4you, 2022

A programação do ESP32 é permitida através de diversos softwares compatíveis, evidenciando-se a linguagem de programação C/C ++, que pode ser desenvolvida através do programa fornecido pela própria desenvolvedora do microcontrolador ou então pelo Arduino IDE.

2.6.2 Software Arduino

O Arduino *Integrated Development Environment* (IDE), é uma plataforma *open source* de desenvolvimento de códigos, onde a partir de um editor de texto é possível descrevê-lo. Este software pode se conectar com o hardware do Arduino e então fazer o upload de programas e se comunicar com eles. Na Figura 13 pode-se observar a interface inicial da plataforma IDE.

Figura 13 – Plataforma IDE



Fonte: Lousada, 2022

A IDE do Arduino utiliza linguagem de programação baseada em C/C++ e possui um grau de abstração muito alto e uma série de bibliotecas que encapsulam a maior parte da complexidade do microcontrolador.

2.7 ACELERÔMETRO

Acelerômetros são dispositivos eletromecânicos capazes de medir forças de aceleração. Essas forças podem ser estáticas, como força da gravidade ou dinâmicas, causadas pela movimentação ou vibração do acelerômetro (LABORATÓRIO IMOBILIS, 2013).

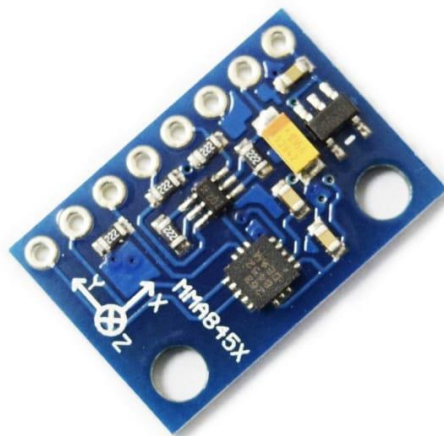
Existem diversos tipos de acelerômetros, alguns usam o efeito piezoelétrico, que contém estruturas microscópicas de cristal que são afetadas pelas forças que atuam no sensor, as quais geram uma tensão elétrica de acordo com a agitação. Outro tipo, é detectando variações a partir da capacitância em duas microestruturas eletrônicas próximas umas das outras, sendo que se a força aplicada for movimentar

as estruturas, a capacitância irá modificar. O fator constante que existe nos elementos eletrônicos é a mudança da tensão de saída proporcionalmente a aplicação sobre o sensor (LABORATÓRIO IMOBILIS, 2013).

2.7.1 Acelerômetro 3 eixos MMA8452

Um acelerômetro de 3 eixos é utilizado para medir a aceleração dinâmica, ou seja, a vibração e a aceleração estática, ou seja, a gravidade. É um dispositivo de 12 bits de resolução com baixo consumo de energia e utiliza a interface I2C para comunicação (FILIPEFLOP, 2022). Na Figura 14 está representada uma imagem do acelerômetro.

Figura 14 – Acelerômetro 3 eixos MMA8452



Fonte: FilipeFlop, 2022

A comunicação I2C é basicamente um circuito integrado para aplicações, os sensores usados no sistema de controle utilizam o protocolo de comunicação de barramento para conectar dispositivos com apenas dois barramentos.

2.8 NODE.JS

O *JavaScript* é uma das linguagens de programação mais populares do mundo. Milhões de websites são alimentados por ele hoje, e atrai diversos desenvolvedores e designers para construir recursos para a web (KINSTA, 2022).

O *Node JS* é um ambiente de tempo de execução com uma camada única, de código aberto e multiplataforma para construir aplicações rápidas e escalonáveis no lado do servidor e em rede (KINSTA, 2022).

2.8.1 *Node RED*

A *Node-RED* é uma ferramenta de programação baseada em fluxo, originalmente desenvolvida pela equipe de Serviços de Tecnologia Emergente da IBM, que agora faz parte da *OpenJS Foundation* (OPENJS FOUNDATION & CONTRIBUTORS, 2022).

A programação baseada em fluxo foi inventada por J. Paul Morrison na década de 1970, sendo uma maneira de descrever o comportamento de um aplicativo como uma rede de caixas pretas, ou “nós”, como chamamos no Node-RED. Cada nó possui um propósito bem definido, receber alguns dados, fazer algo com esses dados e depois repassar esses dados. A rede é responsável pelo fluxo de dados entre os “nós” (OPENJS FOUNDATION & CONTRIBUTORS, 2022).

2.9 *ORANGE DATA MINING*

Orange Data Mining é uma ferramenta *open source* que permite criar todo fluxo de trabalho de um projeto *data mining*, sem necessidade de código. É ideal para práticas de *machine learning*, onde não se faz necessário codificar e é ideal para iniciantes que desejam aplicar alguns conceitos (BATISTA, 2019).

O *Orange* é um *software* de mineração de dados baseado em componentes. Inclui uma variedade de técnicas de visualização, exploração, pré-processamento e modelagem de dados (SAMPAIO, 2020)

3 METODOLOGIA

3.1 MÉTODOS E TÉCNICAS UTILIZADOS

O presente projeto foi desenvolvido em uma indústria metalúrgica focada no ramo de prestação de serviços e produção de itens usinados, localizada no município de Horizontina na região noroeste do estado do Rio Grande do Sul.

O estudo do caso foi caracterizado pelo método quantitativo e de pesquisa descritiva-exploratória. Os procedimentos adotados para a realização da pesquisa foram através de pesquisas bibliográficas e descritivas sobre os processos de usinagem e as principais falhas nas ferramentas de corte, onde usou-se de vários tipos de referências sobre os assuntos abordados, a fim de obter conhecimento sobre eles. O estudo experimental por meio da realização de testes comparativos entre as ferramentas de corte utilizadas na usinagem e manipulação de diversas variáveis que se relacionam com o estudo, como vibração mecânica da ferramenta, temperatura da ferramenta e potência do motor do centro de torneamento.

A partir da metodologia apresentada inicia-se o levantamento dos materiais e equipamentos que serão utilizados na montagem

A técnica de coleta de dados consiste no levantamento e na análise do conteúdo a ser estudado, a partir da extração dos dados de comportamento da ferramenta no centro de torneamento com a utilização de sensores e após observação dos dados retirados a partir do *machine learning*. Na Figura 15 está representado um fluxograma de etapas para a aplicação do *machine learning*.

Figura 15 – Fluxograma de etapas da aplicação do *machine learning*



Na primeira etapa do processo de *machine learning* é necessário entender o problema e definir quais os objetivos do mesmo, no caso deste projeto foi definido que o problema a ser resolvido é a quebra de ferramentas nos processos de usinagem, objetivando a análise dos dados de vibração do motor de um torno CNC, e verificando a possibilidade de prever o desgaste destas ferramentas antes da quebra.

A partir da definição do problema, é necessário definir quais informações são necessárias para realizar a coleta dos dados, para posteriormente serem analisados. Neste caso as informações escolhidas foram as de vibração de um motor, como comentado anteriormente, sendo assim serão coletadas a partir de um protótipo elaborado com um microcontrolador e um sensor de vibração e posteriormente analisadas.

O tratamento dos dados foi feito a partir de um *software* de *machine learning*, onde ele irá construir um padrão de dados e um modelo preditivo, que levarão a solução da proposta solicitada. Após isso, será possível avaliar se o modelo desenvolvido resolverá o problema proposto, a partir da apresentação dos resultados obtidos. Com os resultados em mãos é possível realizar planos de ação para resolver de fato o problema.

3.2 MATERIAIS E EQUIPAMENTOS

Neste processo de pesquisa experimental e coleta de dados, serão analisados os resultados obtidos a fim de realizar a medição das vibrações mecânicas e identificar a partir da utilização de *machine learning* o desgaste de ferramentas durante a usinagem, para que a partir destes dados seja possível tomar ações de manutenções preditivas.

Para a elaboração deste projeto foram utilizados *softwares* e *hardwares*, sendo estes descritos a seguir.

- Arduino IDE;
- *Node JS*;
- *Node RED*;
- *Orange Data Mining*;
- Microcontrolador ESP32;
- Acelerômetro 3 eixos MMA8452;
- Cabos Jumper;
- *Power Bank* 10000mAh (fonte alimentação).

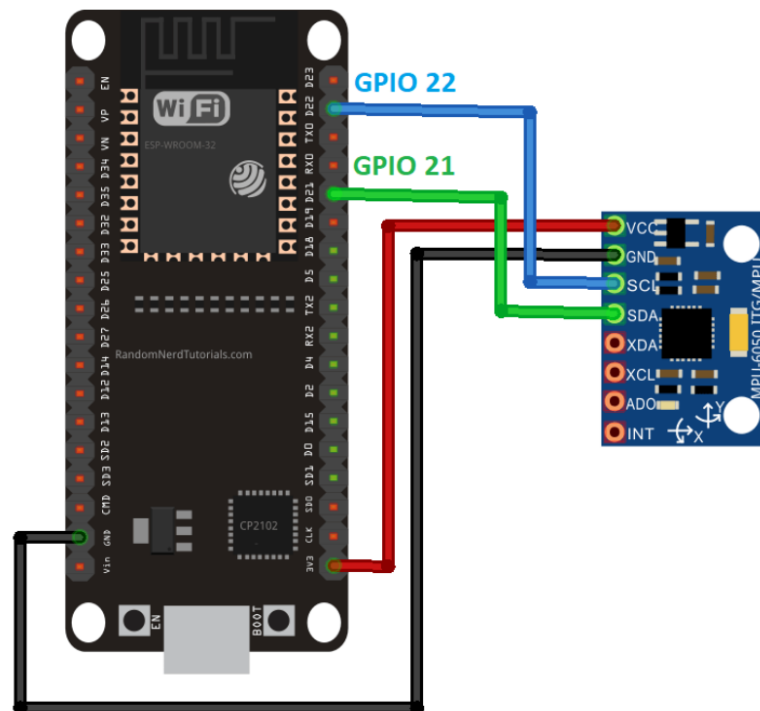
4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Neste tópico são apresentados os resultados obtidos através da metodologia relatada no capítulo anterior.

4.1 SIMULAÇÃO E MONTAGEM DO PROTÓTIPO

Antes de partir para o desenvolvimento do protótipo, foi realizada uma simulação do circuito eletrônico a fim de realizar testes, utilizando o microcontrolador ESP32, cabos e o acelerômetro MMA8452. Na Figura 16 está representado o esquema eletrônico utilizado como base para o desenvolvimento dele.

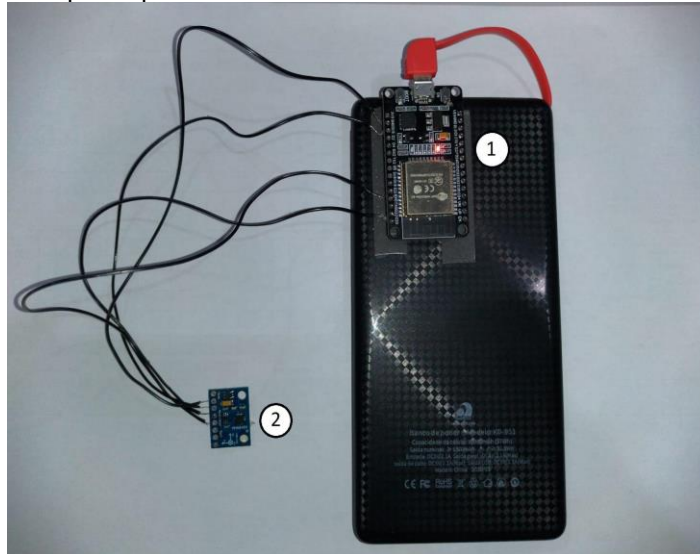
Figura 16 – Esquema eletrônico



Fonte: Random Nerds Tutorials, 2022

Através do esquema elétrico, pode-se realizar a montagem e o teste do protótipo, garantindo sua funcionalidade. A montagem foi realizada conforme a Figura 17. Nela pode-se observar o microcontrolador ESP32 (identificado com o número 1), o acelerômetro (identificado com o número 2), os cabos que fazem a ligação entre os dispositivos e o *power bank* que gera energia para todos os componentes.

Figura 17 – Montagem do protótipo



Fonte: Autor, 2022

Após a montagem e testes do protótipo, ele foi instalado na máquina para realizar o estudo e levantamento dos dados de vibração.

4.2 PROGRAMAÇÃO

A programação do protótipo para realizar a leitura dos dados do acelerômetro foi realizada pela IDE do Arduino e inserida por um cabo USB para o microcontrolador ESP32. O objetivo da programação é coletar os dados do sensor, e enviar para um servidor web, de acordo com o Apêndice.

Na primeira parte da programação foi realizada a comunicação entre o ESP32 e a rede *wi-fi* do local onde foi realizado o estudo. A programação usada como base foi o exemplo "*Simple Web Server WiFi*" fornecida pela própria biblioteca do Arduino, e feito então as alterações necessárias para o funcionamento correto dele. Conforme a Figura 18, a primeira modificação que precisa ser feita é a de informar o nome e a senha da rede na qual o microcontrolador fica conectado.

Figura 18 – Conexão com a rede *wi-fi* pelo Arduino

```

#include <WiFi.h>
#include <Wire.h>

// MMA8452Q I2C address is 0x1C(28)
#define Addr 0x1C

const char* ssid      = "Qualidade";
const char* password = "35376133";

WiFiServer server(80);

```

Fonte: Autor, 2022

Após isso, é necessário alterar o código conforme os comandos no qual deseje-se executar. Sendo assim, na programação foi adicionado o comando para realizar a leitura dos dados enviados pelo sensor, conforme Figura 19.

Figura 19 – Leitura dos dados enviados pelo sensor

```

int value = 0;

void loop() {

    unsigned int data[7];

    // Request 7 bytes of data
    Wire.requestFrom(Addr, 7);

    // Read 7 bytes of data
    // staus, xAccl lsb, xAccl msb, yAccl lsb, yAccl msb, zAccl lsb, zAccl msb
    if (Wire.available() == 7)
    {
        data[0] = Wire.read();
        data[1] = Wire.read();
        data[2] = Wire.read();
        data[3] = Wire.read();
        data[4] = Wire.read();
        data[5] = Wire.read();
        data[6] = Wire.read();
    }
}

```

Fonte: Autor, 2022

Após realizar a leitura dos dados, é necessário realizar o tratamento dos sinais enviados pelo sensor.

4.2.1 Tratamento e processamento dos sinais de entrada

O tratamento e processamento dos sinais de entrada são feitos a partir da transformação dos dados de entrada de 7 bits para os dados de saída em 12 bits. Conforme a Figura 20, é realizado um cálculo padrão para realizar a conversão desses dados.

Figura 20 – Conversão de dados de 7 bits para 12 bits

```
// Convert the data to 12-bits
int xAccl = ((data[1] * 256) + data[2]) / 16;
if (xAccl > 2047)
{
    xAccl -= 4096;
}

int yAccl = ((data[3] * 256) + data[4]) / 16;
if (yAccl > 2047)
{
    yAccl -= 4096;
}

int zAccl = ((data[5] * 256) + data[6]) / 16;
if (zAccl > 2047)
{
    zAccl -= 4096;
}
```

Fonte: Autor, 2022

Após isso, é necessário converter esses dados de bits para aceleração (m/s²), conforme a relação de escala que o Quadro 1 apresenta:

Quadro 1 – Relação de escala aceleração

Modelos	Calibração
2 g	16,384
4 g	8,192
8 g	4,096
16 g	2,048

Fonte: Autor, 2022

Neste projeto, a escala selecionada para o coeficiente de calibração é de 2g. Após definir o ajuste de calibração, deve-se converter os dados conforme a equação 1, no qual define a resolução do valor da gravidade sobre o valor de calibração.

$$Resolução = \frac{9,81}{calibração} \quad (1)$$

Em seguida, a solução encontrada é multiplicada pelo valor lido durante a leitura, definindo assim a aceleração. A Equação 2, exemplifica este cálculo:

$$a = Resolução \times Valor Lido \quad (2)$$

Esse cálculo é feito durante a programação, e o resultado gerado pode ser visualizado no monitor serial do Arduino. Na Figura 21 estão representadas estas variáveis:

Figura 21 – Definição das variáveis

```
float g = 9.8; // Definição do valor da gravidadee  
float resolucao = 0.000604043; // valor da resolução de calibração
```

Fonte: Autor, 2022

Na Figura 22 estão representados os cálculos feitos para calibração do sensor:

Figura 22 – Cálculos de calibração do sensor

```
float gX = resolucao*xAccl;
float gY = resolucao*yAccl;
float gZ = resolucao*zAccl;

// Output data to serial monitor
Serial.print("Acceleration in X-Axis : ");
Serial.println(gX);
Serial.print("Acceleration in Y-Axis : ");
Serial.println(gY);
Serial.print("Acceleration in Z-Axis : ");
Serial.println(gZ-g);
Serial.println(WiFi.localIP());
```

Fonte: Autor, 2022

Após obter os dados do sensor, foi definido que os mesmos seriam armazenados em um servidor web.

4.3 SERVIDOR WEB

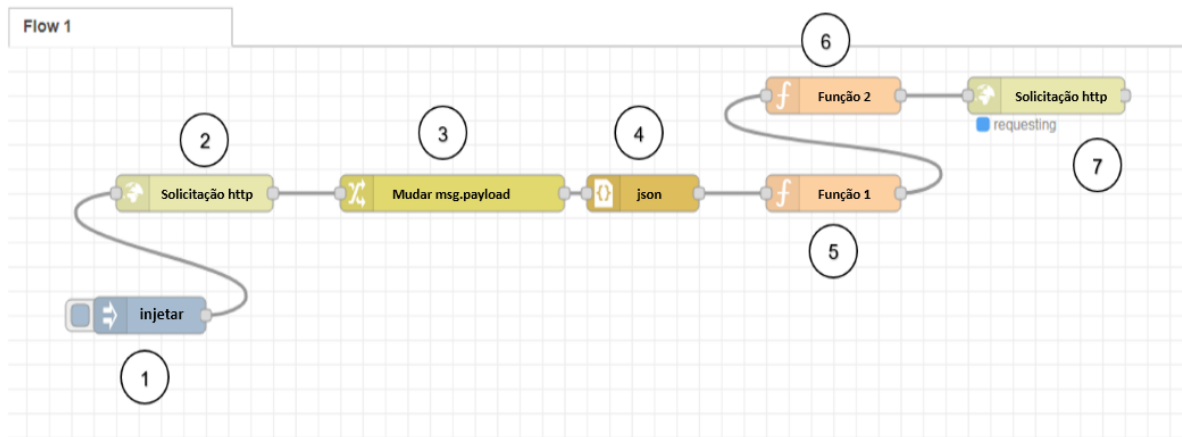
O servidor web foi desenvolvido pelo *software Node JS* e a extensão NPM *Node-RED*, sendo sua principal função receber os dados enviados pelo acelerômetro e armazená-los em um banco de dados online periodicamente, neste caso o local para armazenagem escolhido foi a planilha *Google Sheets*.

A estrutura do servidor web foi dividida conforme a Figura 23, da seguinte maneira:

- Nó 1: responsável por acionar o fluxo do sistema para dispará-lo automaticamente;
- Nó 2: responsável por fazer a requisição dos dados no ESP32 a partir do IP da rede onde o microcontrolador está conectado;
- Nó 3: responsável por trocar as aspas simples por duplas, para deixar a *string* no formato JSON corretamente. Simplesmente uma troca de caracteres da *string*;
- Nó 4: responsável por converter a *string* em JSON, a partir disso o pacote de dados será interpretado como um JSON e não mais uma *string*;

- Nó 5: responsável por inserir uma função desenvolvida em *javascript*, podendo processar a mensagem e enviar somente os dados desejado para o próximo bloco. Neste caso está enviando a variável definida na programação;
- Nó 6: responsável por inserir uma função novamente, porém nesta etapa está sendo convertida em uma variável para ser impressa na planilha online;
- Nó 7: responsável por definir o local que será armazenada as variáveis lidas pelo ESP32. Neste caso foi definido que o local para armazenagem dos dados é a planilha da Google.

Figura 23 – Estrutura do *servidor Web*



Fonte: Autor, 2022

Na definição do local do nó 7, foi adicionado um *link* com a planilha criada no *Google Sheets*, a fim de alocar os dados e as variáveis lidas no sensor de forma online e acessível para os usuários compartilhados. Conforme Figura 24, pode-se observar a forma que os dados são dispostos após a leitura.

Figura 24 – Exemplo da disposição dos dados após leitura no sensor

	A	B	C	D
1	Carimbo de data/hora	acX	acY	acZ
2	19/10/2022 21:08:22 -95.00			
3	19/10/2022 21:08:27 -47.00			
4	19/10/2022 21:08:32 -87.00			
5	19/10/2022 21:08:37 -156.00			
6	19/10/2022 21:08:42 -98.00			
7	19/10/2022 21:08:47 -99.00			
8	19/10/2022 21:08:52 -149.00			
9	19/10/2022 21:08:57 -85.00			
10	19/10/2022 21:09:02 -103.00			
11	19/10/2022 21:09:07 -97.00			
12	19/10/2022 21:09:12 -98.00			
13	19/10/2022 21:09:17 -99.00			

Fonte: Autor, 2022

A planilha armazena os dados de forma periódica e automática, a cada um segundo de leitura do sensor.

4.4 INSTALAÇÃO DO PROTÓTIPO

O modelo de máquina escolhido para o estudo foi um Centro de Torneamento Ergomat - TNG 32, conforme Figura 25. Esse torno é controlado a partir de comandos numéricos, onde pode-se realizar todos os tipos de processos, como por exemplo, desbaste, rosqueamento, furação profunda, etc. Além disso, ele foi desenvolvido para atender a produção seriada de peças torneadas de média a baixa complexidade.

Figura 25 – Centro de Torneamento Ergomat TNG 32



Fonte: Autor, 2022

Na Figura 26 tem-se os dados técnicos do Centro de Torneamento - TNG 32, nele estão contidas informações relevantes ao seu campo de trabalho, motorização, tipo de comando, etc.

Figura 26 – Dados técnicos do torno

■ Campo de trabalho:		
Curso transversal livre eixo X	mm	300
Curso longitudinal livre eixo Z	mm	175
■ Fuso principal:		
Diâmetro máx. barras	mm	32
■ Acionamento principal:		
Rotação máxima do fuso	rpm	5.000
■ Motorização:		
Potência - contínua / intermitente	kW	3,7 / 4,4
■ Acionamento dos eixos:		
Avanço rápido (X / Z)	m/min	12 / 20
■ Mesa porta-ferramentas tipo gang:		
Nº máximo de ferramentas		7
Secção de ferramentas	mm	12 X 12
■ Calha separadora de peças:		
Diâmetro máximo da peça	mm	32
Comprimento máximo da peça	mm	75
■ Sistema de refrigeração:		
Pressão do líquido refrigerante	bar	2 / 4*
Capacidade do tanque	litros	100
■ Comando numérico:		
Fanuc		0i-TF
■ Peso aproximado		
	kg	1.160
Dados técnicos sujeitos a alterações técnicas sem prévio aviso.		
* Opicional		

Fonte: Ergomat, 2018

A fim de detectar as vibrações mecânicas do torno, o protótipo foi fixado juntamente a tampa defletora do motor elétrico, conforme Figura 27.

Figura 27 – Fixação do sensor no motor



Fonte: Autor, 2022

O sensor foi fixado na tampa defletora pois foi o melhor local encontrado, e de fácil acesso. Caso fosse decidido alocar o sensor mais próximo ao eixo, seria necessário desmontar parte da lateral do torno CNC, além de o sensor não ficar protegido contra possíveis fluídos que a máquina poderia eliminar, o que acarretaria na queima do sensor.

O motor estudado é de 3,7 kW, conforme Figura 26, com rotação máxima do eixo de 5000 rpm. Sendo assim, ele pode ser classificado como um motor de Classe I, com condições de trabalho considerado normal até 1,8 rms. Após os dados serem coletados, foi feita uma análise deles, a fim de detectar a partir do *machine learning* anomalias no processo de usinagem.

4.5 APLICAÇÃO DO MACHINE LEARNING

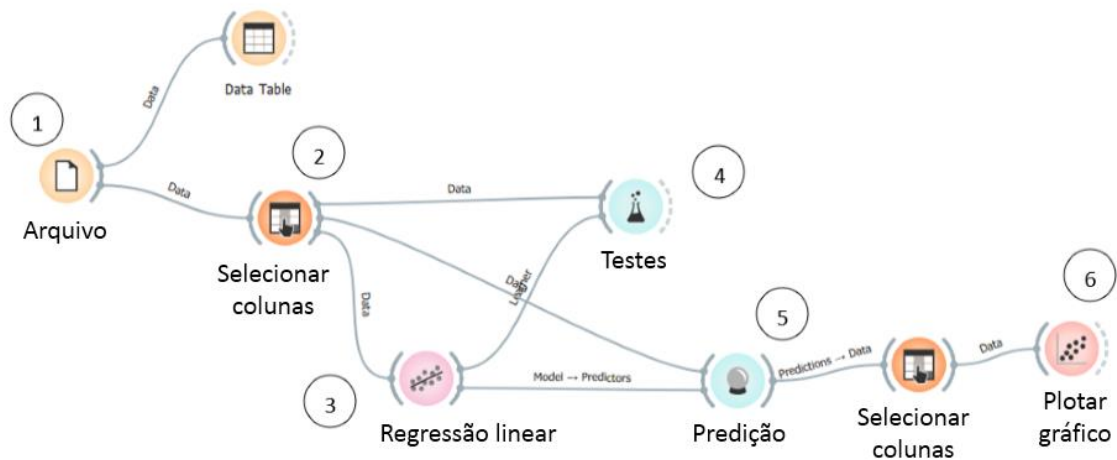
Após os dados serem coletados e armazenados pelo servidor, é necessário realizar a análise deles. Optou-se por utilizar o *software* de *machine learning* *Orange Data Mining*. No *software*, foi criado um fluxo de trabalho inteligente para realizar a análise dos dados extraídos da planilha.

O fluxo de trabalho consiste em selecionar um arquivo em *excel* que contém os dados coletados do sensor (1), após isso, realizar um pré-processamento dos dados, identificando os que serão utilizados como *target* (valores previstos) e *feature* (valores para comparação) (2).

Com os dados selecionados, foi criado então o modelo de análise de *machine learning*. Como foram utilizados dados de aceleração, e os mesmos são considerados valores conhecidos e variam dentro de uma faixa de valor, o modelo indicado e identificado a partir das referências bibliográficas é o de regressão linear, por isso, foi selecionado o modelo deste no *software* (3). Antes de realizar as previsões, foram realizados testes nesse modelo (4), para verificar se ele está avaliando corretamente.

Após verificar que o modelo está avaliando corretamente os dados, foi realizada então a previsão do modelo (5) e plotado o gráfico (6) para análise do mesmo. Na Figura 28 pode-se observar o fluxo de trabalho definido:

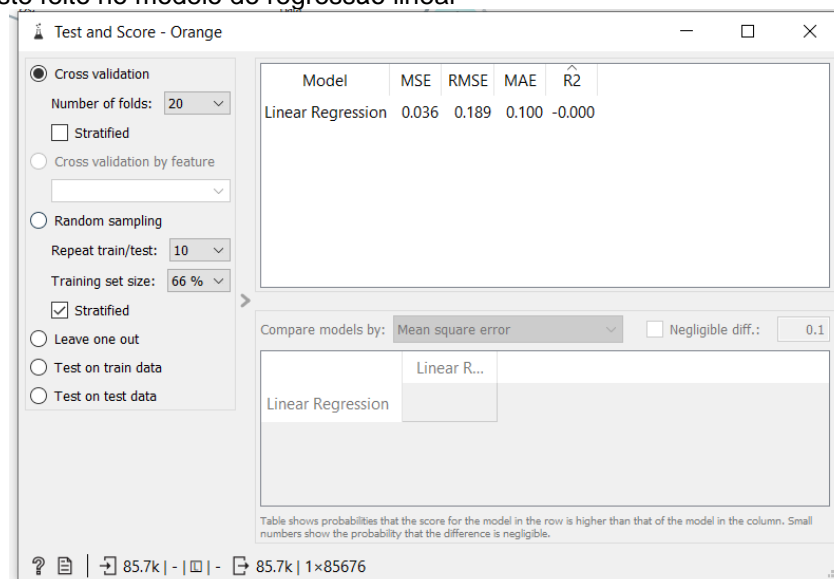
Figura 28 – Fluxo de trabalho no *orange data mining*



Fonte: Autor, 2022

No momento em que foi testado o modelo, pode-se perceber que os resultados esperados não foram satisfatórios. Conforme a Figura 29, os resultados esperados nas colunas MSE, RMSE e MAE deveriam ser próximos de zero, e os da coluna R2 deveria ser próximo de um, porém como verificado, ficou negativo e zerado. Diante disso, o modelo não conseguiu prever corretamente apenas dados com os dados de aceleração coletados.

Figura 29 – Teste feito no modelo de regressão linear



Fonte: Autor, 2022

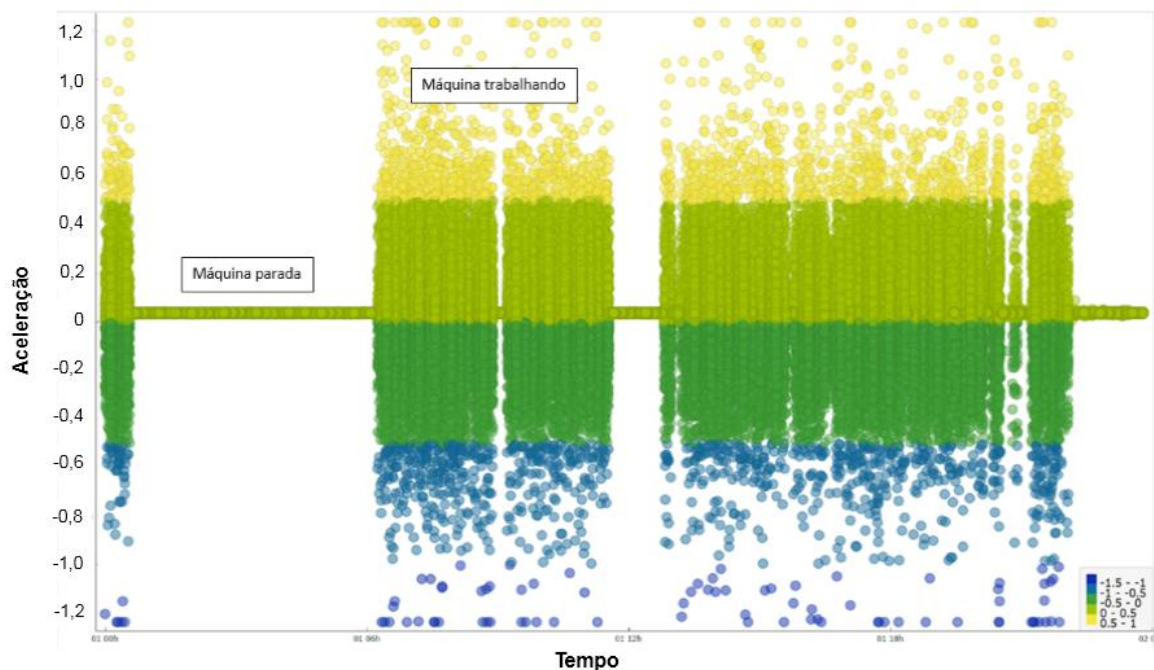
As hipóteses levantadas para o não funcionamento correto do modelo foram, a não coleta de outros dados para realizar a comparação com os dados de vibração e

então criar um padrão de modelo, como por exemplo, de temperatura, umidade e velocidade, no mesmo momento em que estavam sendo coletados os dados de vibração, ou o local inadequado da colocação do sensor na máquina, que deveria ser na carcaça ou próximo ao eixo, ou até mesmo o tipo de sensor utilizado, que poderia ser um industrial, onde teria mais precisão e exatidão.

Conseqüentemente, o modelo não conseguiu prever o momento exato em que deveria ser realizada a troca da ferramenta durante a usinagem, antes de ocorrer a quebra. Porém, com os dados coletados foi possível realizar um monitoramento em tempo real do estado do motor e compará-lo com o estado ideal, conforme a Figura 30.

No gráfico pode-se observar o momento em que o torno CNC está em repouso e no momento em que está trabalhando, sendo que nos momentos em que ela está trabalhando há uma variação normal entre os valores de aceleração em cada momento. Ressaltando que não houve anormalidades durante o processo de coleta de dados, pois os dados estão dentro da faixa de valores de aceleração padrão de motores elétricos (até 1,8 rms).

Figura 30 – Gráfico gerado no *Orange* com os dados de aceleração

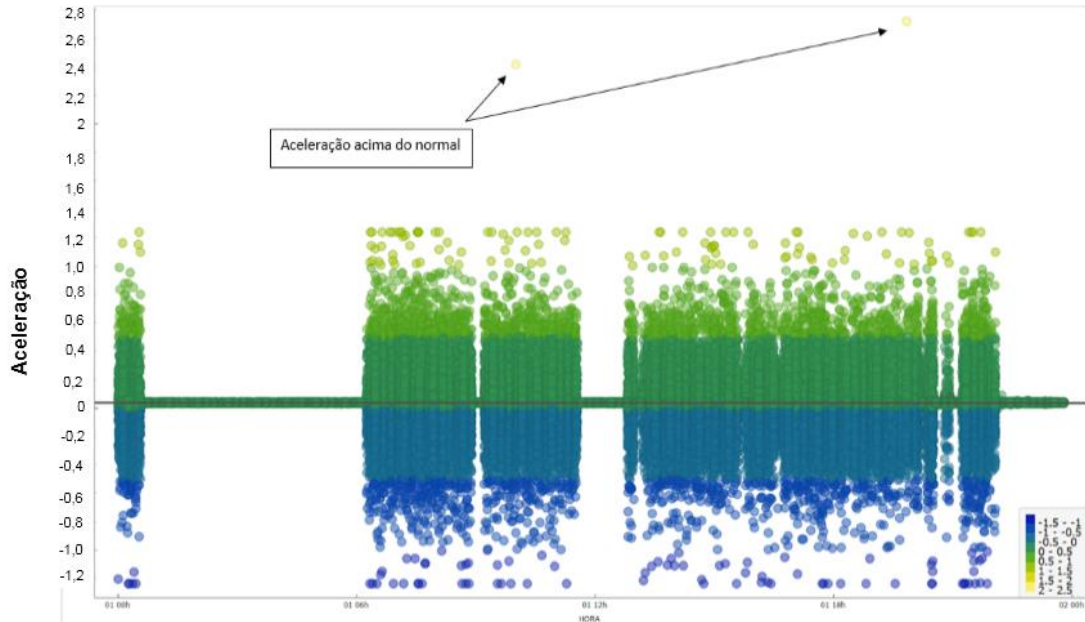


Fonte: Autor, 2022

Para fazer uma comparação entre o estado normal da máquina e o estado anormal, foi alterado manualmente algumas linhas dos dados que foram coletados, adicionando valores que fossem considerados uma aceleração acima ideal, ou seja,

acima 1,8 rms, para que então, fosse possível enxergar no gráfico essa diferença, conforme Figura 31.

Figura 31 – Gráfico com acelerações acima do normal



Fonte: Autor, 2022

Com o gráfico, foi possível verificar esses valores identificados acima da aceleração considerada normal. Sendo assim, nesses momentos, poderia ser identificado que a máquina não estava trabalhando corretamente, e que poderia a qualquer momento ocorrer uma falha ou parada.

CONCLUSÃO

Após o desenvolvimento deste trabalho, foi possível monitorar as vibrações mecânicas de um centro de torneamento durante a usinagem, permitindo acompanhar as variáveis de aceleração do motor, que são de extrema importância para o funcionamento do equipamento. Apesar da aplicação não possibilitar determinar o ponto de troca das ferramentas antes da quebra, o monitoramento das vibrações possibilitou acompanhar o estado do motor em tempo real e compará-lo com o padrão de bibliografias.

A partir do monitoramento das vibrações do motor, com o monitoramento em tempo real, é possível evitar paradas desnecessárias de máquina para manutenção. Conseqüentemente poderá haver aumento na produtividade da produção, incluindo uma cultura de manutenção preditiva.

De forma geral os objetivos citados foram parcialmente alcançados, ele é capaz de realizar a leitura dos dados, fazendo o monitoramento das vibrações, sendo possível armazená-los em um servidor web, e posteriormente analisá-los.

Como melhoria para trabalhos futuros, fica como sugestão a inclusão de outros sensores no protótipo como de temperatura, umidade e velocidade para conseguir fazer a comparação com os dados de aceleração e aplicar o *machine learning* corretamente. Além de incluir uma interação com o usuário para que seja possível visualizar os dados lidos e avisá-lo caso haja alguma anormalidade.

No mercado atual existem empresas, como a WEG e a TRACTIAN que possuem dispositivos capazes de realizar um monitoramento online e realizar a gestão de ativos de uma empresa, fazendo com que as falhas em máquinas sejam evitadas e a manutenção preditiva seja aplicada.

REFERÊNCIAS

- ALVES, Sylvio. **Otimizando processos com Machine Learning na manufatura**. [S. l.]: Medium, 6 jul. 2019. Disponível em: <https://medium.com/@sylvioalves/otimizando-processos-com-machine-learning-na-manufatura-d02a8c454e2c>. Acesso em: 21 mar. 2022.
- BATISTA, Bruno. **Machine Learning sem código: usando Orange Data Mining para criar um modelo preditivo sem usar uma linha de código**. [S. l.]: Medium, 9 jul. 2019. Disponível em: <https://medium.com/ensina-ai/machine-learning-sem-c%C3%B3digo-636d1a8f9081>. Acesso em: 7 nov. 2022.
- BORLIDO, David José Araújo. **Indústria 4.0: aplicação a sistemas de manutenção**. Orientador: Dr. Armando Leitão. 2017. Dissertação (Mestrado Integrado em Engenharia Mecânica) - Faculdade de Engenharia da Universidade do Porto, Portugal, 2017. 65 p. Disponível em: <https://repositorio-aberto.up.pt/bitstream/10216/102740/2/181981.pdf>. Acesso em: 4 abr. 2022.
- CIRCUITS4YOU. **ESP32 DevKit EDP32 - WROOM GPIO Pinout**. [S. l.]: Circuits4you, 31 dez. 2018. Disponível em: <https://circuits4you.com/2018/12/31/esp32-devkit-esp32-wroom-gpio-pinout/>. Acesso em: 19 out. 2022.
- COELHO, Pedro Miguel Nogueira. Rumo a Indústria 4.0. Orientador: Professor Doutor Cristóvão Silva. 2016. Dissertação (Mestre em Engenharia e gestão Industrial) - Faculdade de ciências e tecnologia - Universidade de Coimbra, Coimbra, 2016. p. 62. Disponível em: <http://hdl.handle.net/10316/36992>. Acesso em: 7 set. 2022.
- COSTA, Euder Alves. **Sistema de monitoramento remoto em motores de indução**. Orientador: Prof. Me. Eduardo Henrique Ferroni. 2021. 73 p. Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica) - Centro Universitário do Sul de Minas - UNIS, Varginha/MG, 2021. Disponível em: <http://repositorio.unis.edu.br/handle/prefix/2204>. Acesso em: 3 nov. 2022.
- DAMACENO, Siuari Santos; VASCONCELOS, Rafael Oliveira. Inteligência artificial: uma breve abordagem sobre seu conceito real e o conhecimento popular. **Cadernos de graduação: Ciências exatas e tecnológicas**, Aracaju, ano 1, v. 5, p. 11-16, 1 out. 2018. Disponível em: <https://periodicos.set.edu.br/cadernoexatas/article/view/5729>. Acesso em: 14 mar. 2022.
- DIEZ-OLIVAN, A.; SER, J. D.; GALAR, D.; SIERRA, B. Data fusion and machine learning for industrial prognosis: trends and perspectives towards Industry 4.0. **Information Fusion**, v. 50, p. 92- 111, 2019.
- ERGOMAT. **TNG 32: torno automático CNC**. 8. ed. [S. l.: s. n.], 2018. 2 p. Disponível em: <https://www.ergomat.com.br/torno-cnc-tng-32/>. Acesso em: 1 nov. 2022.
- ESCOVEDO, Tatiana; KOSHIYAMA, Adriano. **Introdução a Data Science: algoritmos de Machine Learning e métodos de análise**. [S. l.]: Casa do Código, 2020. 288 p. ISBN 9788572540551. E-book.

FERRARESI, Dino. **Fundamentos da usinagem dos metais**. São Carlos: Blucher, 1970. 751 p. v. 1.

FILIFELOP. **Acelerômetro 3 eixos MMA8452**. [S. l.]: FilipeFlop, 2022. Disponível em: <https://www.filipeflop.com/produto/acelerometro-3-eixos-mma8452/>. Acesso em: 11 out. 2022.

FRANCISCO, Rodrigo Barros. **Manutenção preditiva na indústria 4.0: a manutenção preditiva como pilar da Indústria 4.0**. Espírito Santo: LinkedIn, 27 set. 2017. Disponível em: <https://www.linkedin.com/pulse/manuten%C3%A7%C3%A3o-preditiva-na-ind%C3%BAstria-40-rodrigo-barros-francisco/>. Acesso em: 11 abr. 2022.

GONÇALVES, Pedro Henrique Cantelli. **Aplicação de ferramenta de machine learning para predição de falhas em equipamentos em indústria de papel**. Orientador: Prof. Dr. Rui Tadashi Yoshino. 2020. Trabalho de conclusão de curso (Especialista em indústria 4.0) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2020. 13. p. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/25960>. Acesso em: 19 mar. 2022.

HEIDENREICH, Hunter. **What are the types of machine learning?** [S. l.]: Towards Data Science, 4 dez. 2018. Disponível em: <https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f>. Acesso em: 4 abr. 2022.

HONDA, Hugo; FACURE, Matheus; YAOHAO, Peng. **Os três tipos de aprendizado de máquina**. [S. l.]: LAMFO, 27 jul. 2017. Disponível em: <https://lamfo-unb.github.io/2017/07/27/tres-tipos-am/>. Acesso em: 4 abr. 2022.

HORN, André Luiz. **Indústria 4.0 melhorando a eficiência na manufatura**. Orientador: Marcelo Gechele Cleto. 2019. 21 p. Trabalho de conclusão de curso (Pós Graduação em Engenharia de Produção) - Universidade Federal do Paraná, Curitiba, 2019. Disponível em: <https://hdl.handle.net/1884/70361>. Acesso em: 1 mar. 2022.

IBRAHIM, Dogan. **The Official ESP32 Book**. [S. l.]: Elektor International Media, 2017. 286 p. ISBN 1907920633.

ISO 20816-1/ISO. Abstract to ISO 20816-1. ISO, 2016. Disponível em: <https://www.iso.org/standard/63180.html>. Acesso em: 4 nov. 2022.

JUNIOR, José S. **Introdução às vibrações mecânicas**. São Paulo: Editora Blucher, 2006. E-book. ISBN 9788521214953. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521214953/>. Acesso em: 30 nov. 2022.

KENJI, Bruno. **Machine Learning para leigos**. [S. l.]: Venturus, 18 dez. 2019. Disponível em: <https://www.venturus.org.br/machine-learning-para-leigos/>. Acesso em: 4 abr. 2022.

KINSTA. **WHAT Is Node.js and Why You Should Use It**. [S. l.]: KINSTA, 8 jun. 2022. Disponível em: <https://kinsta.com/knowledgebase/what-is-node-js/>. Acesso em: 13 set. 2022.

KOLBAN, Neil. **KOLBAN 'S Book on ESP32**. [S. l.: s. n.], 2018. Disponível em: <https://leanpub.com/kolban-ESP32>. Acesso em: 7 nov. 2022.

LABORATÓRIO IMOBILIS. **Acelerômetros: uso em celulares e detecção de velocidade. parte 1**. [S. l.]: Laboratório Mobilis, 30 jan. 2013. Disponível em: <http://www2.decom.ufop.br/imobilis/acelerometros-uso-em-celulares-e-deteccao-de-velocidade-parte-1/#:~:text=Um%20aceler%C3%B4metro%20%C3%A9%20um%20dispositivo,movimenta%C3%A7%C3%A3o%20ou%20vibra%C3%A7%C3%A3o%20do%20aceler%C3%B4metro>. Acesso em: 18 out. 2022.

LINDSTRÖM, J.; LEJON, E.; KYÖSTI, P.; MECELLA, M.; HEUTELBECK, D.; HEMMJE, M.; SJÖDAHL, M.; BIRK, W.; GUNNARSSON, B. (2019). Towards intelligent and sustainable production systems with a zero-defect manufacturing approach in an Industry4.0. **Context. Procedia CIRP**, v. 81, p. 880- 885, 2019.

LOUSADA, Ricardo. **O que é Arduino: para que serve, vantagens e como utilizar**. [S. l.]: Blog Eletrogate, 8 set. 2022. Disponível em: <https://blog.eletrogate.com/o-que-e-arduino-para-que-serve-vantagens-e-como-utilizar/>. Acesso em: 19 out. 2022.

MACHADO, Álisson R.; COELHO, Reginaldo T.; ABRÃO, Alexandre M. **Teoria da usinagem dos materiais**. São Paulo: Editora Blucher, 2015. E-book. ISBN 9788521208440. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788521208440/>. Acesso em: 29 nov. 2022.

MARCORIN, Wilson Roberto; LIMA, Carlos Roberto Camello. Análise dos custos de manutenção e de não-manutenção de equipamentos produtivos. **Revista de Ciência & Tecnologia**, Piracicaba, v. 11, n. 22, p. 35-42, 18 dez. 2003. Disponível em: <http://www.drb-m.org/av1/11Custodemantencao.pdf>. Acesso em: 23 mar. 2022.

DOS NETO, Antonio dos Reis de F.; ASSUNÇÃO, Germano S C.; SANTOS, Felipe D. Dias; et al. **Vibrações Mecânicas**. Porto Alegre: Grupo A, 2022. E-book. ISBN 9786556902654. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556902654/>. Acesso em: 29 nov. 2022.

OPENJS FOUNDATION & CONTRIBUTORS. **NODE-RED**. [S. l.], 2022. Disponível em: <https://nodered.org>. Acesso em: 3 out. 2022.

PERES, Fabrina Barbieri; UEMURA, Larissa Caliani; ZANGHETTIN, Larissa. Energia mecânica com ênfase em processos de usinagem: Torneamento. REGRAD: **UNIVEM**, Marília - SP, v. 10, n. 1, p. 325-338, 2017. Disponível em: <https://revista.univem.edu.br/REGRAD/article/view/2062/679>. Acesso em: 4 abr. 2022.

PINTO, Alan Kardec e XAVIER, Júlio Nascif. **Manutenção: função estratégica**. Rio de Janeiro: Qualitymark, 2007.

PREEZ, Anli du; OOSTHUIZEN, Gert Adriaan. Machine learning in cutting processes as enabler for smart sustainable manufacturing. **Science Direct**, [s. l.], p. 210-217, 2019. DOI <https://doi.org/10.1016/j.promfg.2019.04.102>. Disponível em:

<https://www.sciencedirect.com/science/article/pii/S2351978919305839>. Acesso em: 22 mar. 2022.

RANDOM NERDS TUTORIALS. **ESP32 with MPU-6050 Accelerometer, Gyroscope and Temperature Sensor (Arduino)**. [S. l.]: Random, 2022. Disponível em: <https://randomnerdtutorials.com/esp32-mpu-6050-accelerometer-gyroscope-arduino/>. Acesso em: 7 nov. 2022.

SACOMANO, José Benedito; GONÇALVES, Rodrigo Franco; BONILLA, Sílvia Helena; DA SILVA, Márcia Terra; SÁTYRO, Walter Cardoso. **Indústria 4.0**. [S. l.]: Blucher, 2018. 169 p.

SAMPAIO, Rogério Silva. **Utilizando o Orange 3 para avaliação de modelos de Machine Learning**. 29 dez. 2020. LinkedIn: Rogerio Silva Sampaio. Disponível em: <https://pt.linkedin.com/pulse/utilizando-o-orange-3-para-avalia%C3%A7%C3%A3o-de-modelos-rogerio-silva-sampaio>. Acesso em: 11 out. 2022.

SANTOS, Sandro Cardoso; SALES, Wisley Falco. **Aspectos tribológicos da usinagem dos metais**. São Paulo: Artliber, 2007. 220 p.

SCHWAB, Klaus. **A quarta revolução industrial**. [S. l.]: Edipro, 2019.

SILVA, Fernanda R.; SOARES, Juliane A.; SERPA, Matheus da S.; et al. **Cloud Computing**. Porto Alegre: Grupo A, 2020. E-book. ISBN 9786556900193. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556900193/>. Acesso em: 29 nov. 2022.

SILVEIRA, Cristiano Bertulucci. **Indústria 4.0: o que é, e como ela vai impactar o mundo**. São Paulo: Citisystems, 11 fev. 2016. Disponível em: <https://www.citisystems.com.br/industria-4-0/>. Acesso em: 4 abr. 2022.

SOUZA, André João de. **Monitoramento de processos de usinagem**. Porto Alegre: [s. n.], 2015. 54 p.

VELASQUEZ, Luis Henrique. **Uma visão geral sobre machine learning: classificação**. [S. l.]: StatPlace, 2022. Disponível em: [https://statplace.com.br/blog/uma-visao-geral-sobre-machine-learning/#:~:text=Classifica%C3%A7%C3%A3o%20no%20Machine%20Learning,\(tr%C3%AAs%20ou%20mais%20classes\)](https://statplace.com.br/blog/uma-visao-geral-sobre-machine-learning/#:~:text=Classifica%C3%A7%C3%A3o%20no%20Machine%20Learning,(tr%C3%AAs%20ou%20mais%20classes)). Acesso em: 4 nov. 2022.

VELLOSO, Higor Medina; HORA, Henrique Rego Monteiro da. Classificação de falhas de um centro de usinagem: um estudo de caso utilizando árvore de decisão. SIMPÓSIO DE PESQUISA OPERACIONAL & LOGÍSTICA DA MARINHA, 19, São Paulo, ano 2020, v. 3, p. 1-9, 1 maio 2020. DOI 10.5151/spolm2019-072. Disponível em: <https://www.proceedings.blucher.com.br/article-details/classificao-de-falhas-de-um-centro-de-usinagem-um-estudo-de-caso-utilizando-rvore-de-deciso-34487>. Acesso em: 24 fev. 2022.

VIEIRA, Hugo Fernandes. **Detecção de fraude no uso de cartões de crédito utilizando técnicas supervisionadas de machine learning**. Orientador: Prof. Me. Marcelo de Castro Cardoso. 2018. Trabalho de conclusão de curso (Bacharelado em Engenharia de Computação) - Centro Universitário de Anápolis, Anápolis, 2018. p. 51.

Disponível em: http://45.4.96.19/bitstream/aee/9334/1/2_MaHugo.pdf. Acesso em: 7 abr. 2022.

VON LUXBURG, U.; SCHÖLKOPF, B. Statistical learning theory: models, concepts, and results. **ArXiv preprint arXiv:0810.4752**, n. 1, oct. 2008.

APÊNDICE

```
#include <WiFi.h>

#include <Wire.h>

// MMA8452Q I2C address is 0x1C(28)

#define Addr 0x1C

const char* ssid    = "Qualidade";
const char* password = "35376133";

float g = 9.8; // Definição do valor da gravidade
float resolucao = 0.000604043; // valor da resolução de calibração

WiFiServer server(80);

void setup()
{
  Serial.begin(115200);
  pinMode(34, INPUT);
  pinMode(35, INPUT);
  pinMode(25, INPUT);
  pinMode(15, OUTPUT);
  pinMode(32, OUTPUT);

  delay(10);
  digitalWrite(15, LOW);
  digitalWrite(32, LOW);

  // We start by connecting to a WiFi network
```

```
Serial.println();  
Serial.println();  
Serial.print("Connecting to ");  
Serial.println(ssid);  
WiFi.begin(ssid, password);  
  
int contador = 0;  
  
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
    contador++;  
  
    Serial.println("");  
    Serial.println("WiFi connected.");  
    Serial.println("IP address: ");  
  
    if (contador >= 10) {  
        ESP.restart();  
    }  
}
```



```
server.begin();

// Initialise I2C communication as MASTER
Wire.begin();
// Initialise Serial Communication, set baud rate = 9600
//Serial.begin(9600);

// Start I2C Transmission
Wire.beginTransmission(Addr);
// Select control register
Wire.write(0x2A);
// StandBy mode
Wire.write((byte)0x00);
// Stop I2C Transmission
Wire.endTransmission();

// Start I2C Transmission
Wire.beginTransmission(Addr);
// Select control register
Wire.write(0x2A);
// Active mode
Wire.write(0x01);
// Stop I2C Transmission
Wire.endTransmission();

// Start I2C Transmission
Wire.beginTransmission(Addr);
```

```
// Select control register
Wire.write(0x0E);
// Set range to +/- 2g
Wire.write((byte)0x00);
// Stop I2C Transmission
Wire.endTransmission();
delay(300);

}

int value = 0;

void loop() {

    unsigned int data[7];

    // Request 7 bytes of data
    Wire.requestFrom(Addr, 7);

    // Read 7 bytes of data
    // staus, xAccl lsb, xAccl msb, yAccl lsb, yAccl msb, zAccl lsb, zAccl msb
    if (Wire.available() == 7)
    {
        data[0] = Wire.read();
        data[1] = Wire.read();
        data[2] = Wire.read();
        data[3] = Wire.read();
    }
}
```

```
data[4] = Wire.read();
data[5] = Wire.read();
data[6] = Wire.read();
}

// Convert the data to 12-bits
int xAccl = ((data[1] * 256) + data[2]) / 16;
if (xAccl > 2047)
{
    xAccl -= 4096;
}

int yAccl = ((data[3] * 256) + data[4]) / 16;
if (yAccl > 2047)
{
    yAccl -= 4096;
}

int zAccl = ((data[5] * 256) + data[6]) / 16;
if (zAccl > 2047)
{
    zAccl -= 4096;
}

float gX = resolucao*xAccl;
float gY = resolucao*yAccl;
float gZ = resolucao*zAccl;
```

```
// Output data to serial monitor
```

```
Serial.print("Acceleration in X-Axis : ");
```

```
Serial.println(gX);
```

```
Serial.print("Acceleration in Y-Axis : ");
```

```
Serial.println(gY);
```

```
Serial.print("Acceleration in Z-Axis : ");
```

```
Serial.println(gZ-g);
```

```
Serial.println(WiFi.localIP());
```

```
delay(500);
```

```
//=====
=====
```

```
WiFiClient client = server.available(); // listen for incoming clients
```

```
if (client) { // if you get a client,
```

```
    Serial.println("New Client."); // print a message out the serial port
```

```
    String currentLine = ""; // make a String to hold incoming data from the
client
```

```
    while (client.connected()) { // loop while the client's connected
```

```
        if (client.available()) { // if there's bytes to read from the client,
```

```
            char c = client.read(); // read a byte, then
```

```
            Serial.write(c); // print it out the serial monitor
```

```
            if (c == '\n') { // if the byte is a newline character
```

```
                if (currentLine.length() == 0) {
```

```
// send a standard http response header
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: application/json");
client.println("Connection: close"); // the connection will be closed after
completion of the response

client.println("{ 'type': 'Vibricao','value':");
client.println(xAccl);
client.println("}],");

client.println(" 'booleanData':[");

client.println("]");
break;
} else {
    currentLine = "";
}
} else if (c != '\r') {
    currentLine += c;
}
}
}
// close the connection:
client.stop();
Serial.println("Client Disconnected.");
}
}
```