Universidade de Aveiro
2018

Departamento de Engenharia Mecânica
Departamento de Eletrónica Telecomunicações e Informática

André Manuel Marques dos Santos

ALGORITMOS PARA A MANUTENÇÃO PREDITIVA, NA INDÚSTRIA 4.0

Universidade de Aveiro
2018

Departamento de Engenharia Mecânica
Departamento de Eletrónica Telecomunicações e Informática

André Manuel Marques dos Santos

ALGORITMOS PARA A MANUTENÇÃO PREDITIVA, NA INDÚSTRIA 4.0

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Automação Industrial, realizada sob a orientação científica do Professor Doutor José Paulo Oliveira Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro e do Professor Abílio Manuel Ribeiro Borges, Assistente Convidado da Universidade de Aveiro.

Colaboração Institucional
The Navigator Company, Navigator Pulp Cacia

Dedico este trabalho a todos aqueles que acreditaram em mim!

O júri

Presidente

Professor Doutor Pedro Nicolau Faria da Fonseca

Professor auxiliar da Universidade de Aveiro

Professor Doutor José Paulo Oliveira Santos

Professor auxiliar da Universidade de Aveiro

Professora Doutora Ana Maria Pinto de Moura

Professora auxiliar da Universidade de Aveiro

agradecimentos

Agradeço de forma geral a todos aqueles que puderam ser pacientes comigo e que me acompanharam ao longo deste percurso académico. Agradeço a minha mãe pela valorização incondicional e a minha irmã que de alguma forma vai estar sempre presente.

Aos meus colegas e amigos que contribuíram com conhecimento, experiência e uma boa disposição.

Ao Professor Doutor José Paulo Oliveira Santos e ao Professor Abílio Manuel Ribeiro Borges pelo apoio e dedicação.

Ao Eng.º. Carlos Manuel da Costa Pinto Gonçalves, Diretor de manutenção do Centro Fabril de Cacia da The Navigator Company, ao Eng.º. Fernando Silva Meireles, encarregado geral da oficina de controlo e potência e ao Eng.º Paulo Armando Carteiro Lopes, Supervisor de manutenção na área de Recuperação e Energia, um grande obrigado pela oportunidade que me deram não só de progredir academicamente, mas também de utilizar o ambiente fabril para testes e medições. Devo agradecer à Navigator SA como entidade por valorizar o interesse dos seus colaboradores, como por exemplo esta oportunidade.

palavras-chave Manutenção Preditiva, Indústria 4.0, Algoritmos Preditivos, Vibrações, Machine Learning, Multi-layer Perceptron Classifier, Rolamentos,

resumo No âmbito da Indústria 4.0, pretende-se o desenvolvimento de algoritmos que permitam realizar a manutenção preditiva e apoiar na tomada de decisões face à calendarização de uma paragem de produção para reparações. Desta forma o planeamento de uma paragem será mais preciso, diminuindo custos, evitando a aquisição de material/equipamentos com entrega imediata (valor acrescido) e o *overstock* em armazém. Este projeto contempla um estudo sobre os vários indicadores que contribuem para a identificação de falha de rolamentos, como por exemplo a temperatura e as frequências inerentes aos constituintes do rolamento. Tem também uma análise de métodos preditivos focada em redes neuronais artificiais, particularmente num algoritmo chamado de *Multi-layer Perceptron Classifier*. Este algoritmo é aplicado sobre vibrações sob a forma de frequência e medições de temperatura. A aceleração e a temperatura, são recebidas em tempo real de pontos estratégicos da máquina, e que demonstrem o estado atual de degradação dos rolamentos. Estas resultam do processamento de medições de aceleração com o algoritmo *Fast computation algorithm for discrete Fourier Transform* sendo as frequências e temperatura submetidas ao modelo que indica qual o tempo de vida que a máquina tem. Foram criadas duas rotinas de atualização do modelo de aprendizagem, uma que se repete semanalmente e outra que ocorre de aproximadamente de 16 em 16 horas.

Por fim são apresentados vários resultados que comprovam a eficiência do algoritmo de *machine-learning.*

keywords Predictive Maintenance, Industry 4.0, Predictive Algorithms, Vibrations, Machine Learning, Multi-layer Perceptron Classifier, Bearings

Abstract

In the scope of Industry 4.0, it is intended the development of algorithms that allow carrying out the predictive maintenance and support in the decision-making in relation to the scheduling of a production stop for repairs. In this way the planning of a stop will be more precise, reducing costs and avoiding urgent purchases and overstock in the warehouse.

This project contemplates a study on the various indicators that contribute to the identification of bearing failure like temperature and each component frequency, and has a somewhat detailed analysis on predictive methods focused on Artificial Neural Network, particularly in an algorithm called Multi-layer Perceptron Classifier.

That algorithm is applied on a quantity of data, vibrations in the form of frequency and measurements of temperature. The data is received in real time from strategic points of the machine that demonstrates the current degradation state of the machine. They are then processed and submitted to the model which then calculates the machine life span. Two update routines of the learning model were created, one that is repeated weekly and the other that occurs approximately 16 to 16 hours. This makes the model approach more reliable, even with load changes.

Finally, several results that demonstrate the efficiency of the machine-learning algorithm are presented and discussed on the last chapter.

Índice

1. Introdução 2
1.1 Contextualização 2
1.2 Problema a resolver e qual a sua importância 2
1.3 Resultados esperados 3
1.4 Organização da dissertação 4
2 A Empresa 6
2.1 História da Empresa 6
2.2 Aplicabilidade e Processo 7
3 Estado da arte 12
3.1 Manutenção na Industria 4.0 12
3.1.1 A internet das coisas na Indústria (IIoT) 13
3.1.2 Big Data 14
3.1.3 Tipos de Manutenção 15
3.2 Algoritmos preditivos 18
3.3 Machine Learning 20
3.3.1 Naive Bayesian 22
3.3.2 Decision Trees e Random Forests 23
3.3.3 Hidden Markov Chains 23
3.3.4 Support Vector Machines 24
3.3.5 Redes Neuronais Artificiais 25
3.4 Análise de vibrações 33
3.5 Influência da Temperatura no tempo de vida de rolamentos 39
3.6 Soluções comerciais atuais 40
4 Implementação e análise de desempenho 42

4.1 Aquisição de dados reais 43
4.2 Programação do microcontrolador e interligação com a base de dados 47
4.3 Interfaces Gráficas 51
4.4 Modelos de Machine Learning e resultados de aprendizagem 56
5 Conclusões 70
Referências 72

Índice de figuras

Figura 2-1 - Navigator Pulp Cacia[1] 6
Figura 2-2 - Processo de fabrico de Papel [2] 7
Figura 2-3 - Esquema simplificado da zona húmida e secagem da Tiragem 4 10
Figura 2-4 - Chumaceira do rolo 4PR24 (Rolo esticador) 10
Figura 3-1 - Evolução da indústria [3] 12
Figura 3-2 - Os 9 pilares da indústria 4.0 13
Figura 3-3 - MySQL vs. Microsoft SQL Server[5] 15
Figura 3-4 - Gráfico temporal do custo para repara uma avaria vs. tipos de manutenção [7] 17
Figura 3-5 - Algoritmo Genérico 18
Figura 3-6 - Algoritmo básico de manutenção 19
Figura 3-7 – Análise de tendência vs. análise de limites 20
Figura 3-8 - Categorias de machine learning e suas aplicações [8] 20
Figura 3-9 - Árvore binária [10] 23
Figura 3-10 – Representação gráfica de SVM [13] 24
Figura 3-11 - Separação de dados por um plano [14] 25
Figura 3-12 - Cérebro como inspiração para as redes neuronais artificiais [44] 25
Figura 3-13 – Exemplo de uma RNA (3,2,1)[10] 27
Figura 3-14 - Gráfico da função sigmoid (azul) e da sua derivada (verde)[16] 28
Figura 3-15 - Rede neuronal de 2-camadas [10] 29
Figura 3-16 - Rede neuronal complexa, função complexa[9] 32
Figura 3-17 - Fast Fourier Transform [20] 35
Figura 3-18 - Gráfico FFT de desequilíbrio (*Unbalance*) e desalinhamento (*Misalignment*) 37
Figura 3-19 - Partes constituintes de um Rolamento [23] 37
Figura 4-1 – Diagrama do Projeto 42
Figura 4-2 – Medição de frequências de vibração de um motor em ambiente industrial 43
Figura 4-3 - Equipamento montado para medição de vibrações 46
Figura 4-4 - Comparação da medição FFT 48
Figura 4-5 - Representação da base de dados 49

Figura 4-6 – Teste para ajuste na determinação de frequências medidas na ADC do ESP8266 50
Figura 4-7 - Medições da frequência do sinal gerado pelo PicoScope 2000 (60 Hz) 51
Figura 4-8 - Programação por flow/nodes 51
Figura 4-9 - Configuração de módulos, à direita por interface gráfica e à esquerda por programação 52
Figura 4-10 – SCADA de monitorização 53
Figura 4-11 - Interface gráfica do programa de configuração do equipamento a medir 54
Figura 4-12 - Configuração manual do rolamento 55
Figura 4-13 - Tipos de algoritmos em scikit-learn[18] 58
Figura 4-14 - Parâmetros do algoritmo de machine learning MLPC 59
Figura 4-15 - Rede Neuronal Artificial Usada 59
Figura 4-16 - Processo de aquisição, tratamento de dados e previsão do tempo de vida dos rolamentos 60
Figura 4-17 - Relatório do resultado da aprendizagem no DOS 61
Figura 4-18 - Algoritmo proposto para Aprendizagem contínua 65
Figura 4-19 - Relatório da aprendizagem para 1 opção de saída 65
Figura 4-20 - Relatório da aprendizagem para 2 opções de saída 66
Figura 4-21 Relatório da aprendizagem para 3 opções de saída 66
Figura 4-22 - Relatórios de várias aprendizagens 67
Figura 4-23 - Percentagem de Erro (528 linhas de teste) 68

Índice de Tabelas

Tabela 3-1 - ISO 2372 Standard.....36
Tabela 4-1 - Características dos motores analisados.....44
Tabela 4-2 – Excerto da tabela de dados reais medidos em ambiente industrial do motor Diag1.....45
Tabela 4-3 - Comparação entre sensores de vibração.....46
Tabela 4-4 - Dados de entrada para o processo de aprendizagem.....56
Tabela 4-5 - Exemplo de matriz de confusão.....61

Lista de Acrónimos

SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung, Sistemas, Aplicações e Produtos para Processamento de Dados
Adam	Adaptive Moment Estimation
ADC	Analog to Digital Converter
ANN/RNA	Artificial Neural Networks / Redes Neuronais Artificiais
API	Aplication Programing Interface
BPFI	Ball Pass Frequency of Inner ring
BPFO	Ball Pass Frequency of Outer ring
BSF/BPF	Ball Spin Frequency
DBMS	Database Management System
DFT	Discrete Fourier Transform
DOS	Disk Operating System
FFT	Fast computation algorithm for discrete Fourier Transform
HMM	Hidden Marcov Machines
HTTPD	Hypertext Transfer Protocol Server Daemon (Apache)
I^2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IIoT	Industrial Internet of Things, Internet das coias no contexto industrial
IoT	Internet of Things, Internet das coisas
IT	Information Technology
LAMP	Linux, Apache, MySQL, PHP
L-BFGS	algoritmo de Broyden–Fletcher–Goldfarb–Shanno com memória Limitada
MLPC	Multi-Layer Perceptron Classifier
MSE	Mean Square Error
MTBF	Mean Time Between Failures
MTBR	Mean Time Between Repairs
PCB	Printed Circuit Board
PHP	Hypertext Preprocessor, originalmente Personal Home Page
RPM	Rotations per minute

SGD	Stochastic Gradient Descent
SMD	Surface-Mount Device
SQL	Structured Query Language
STFT	Short-Time Fourier Transform
SVM	Support Vector Machines
TCO	Total Cost of Ownership
TCP/IP	Trasmission Control Protocol/Internet Protocol
THD	Total Harmonic Distortion
WVD	Wigner – Ville Distribution
XAMPP	X – qualquer sistema operativo, Apache, MySQL, PHP, Perl
ZDT	Zero Down Time

1. Introdução

1.1 Contextualização

A Manutenção Industrial é importante devido aos elevados custos que as avarias e consequentes paragens de produção acarretam. Deste modo, é necessário desenvolver metodologias de apoio à tomada de decisão na Manutenção que permitam uma gestão eficiente da mesma.
Durante muito tempo, as indústrias funcionaram apenas com sistemas de manutenção corretiva, ou seja, realizavam a reparação de peças e equipamentos somente após a ocorrência de falha. Com a globalização dos mercados e consequente concorrência, as empresas vêem-se cada vez mais confrontadas com a necessidade de evoluir, não só através da melhoria da qualidade no seu produto/serviço, mas também na redução de custos. Na perspetiva da redução de custos associados ao tempo fora de serviço é, cada vez mais, necessário desenvolver arquiteturas que tornem as causas de avaria previsíveis para evitar falhas mais graves. A competitividade e a organização das empresas quando atuam na área da manutenção podem ser fortalecidas consideravelmente com a ajuda de uma solução que combine as novas tecnologias existentes aos sistemas já implantados no sentido de prever uma avaria.

1.2 Problema a resolver e qual a sua importância

Pretende-se o desenvolvimento de algoritmos que permitam realizar a manutenção preditiva e apoiar a tomada de decisões. Estes algoritmos serão aplicados sobre uma quantidade de valores de vibração e temperatura medidos em tempo real, em pontos estratégicos e que demonstrem o estado atual de uma máquina. Desta forma o planeamento de uma paragem será mais preciso, diminuindo custos, evitando a aquisição de material/equipamentos com entrega imediata (valor acrescido) e o overstock em armazém.

Hipoteticamente, assumindo que a empresa produz 1000 toneladas de pasta de papel por dia, e que o mercado da pasta de papel ronda os 1000€ por tonelada, numa fábrica, como esta, de laboração

contínua, significa uma faturação de 1 M€ em 24 horas. Com estes valores, tem-se uma ideia de que com uma paragem de apenas 1 hora, significa um custo de oportunidade bastante significativo (≈41000€).

1.3 Resultados esperados

Esta dissertação visa propor uma solução para a manutenção preditiva em 2 locais estratégicos na fábrica. O primeiro é a monitorização do motor e respetivas chumaceiras do ventilador de ar secundário da Caldeira de Recuperação nº 4, que embora não interfira diretamente com o processo de produção de pasta de papel, é um componente vital para a caldeira, consequentemente para a produção de energia, e sem energia, elétrica ou sob a forma de vapor, tudo para. O segundo local é a "Tiragem IV", que sofre bastante desgaste e interrupções ao processo por paragens normalmente originadas por falhas em rolamentos.

Como tal, o espectado é ter vários sensores montados nesses equipamentos, recolher leituras de diferentes grandezas em tempo real e construir uma base de dados. Se possível, adicionalmente, analisar dados provenientes da plataforma SAP (do alemão, ***S****ysteme, **A**nwendungen und **P**rodukte in der Datenverarbeitung* que significa, Sistemas, Aplicações e Produtos para Processamento de Dados) para contribuir com o histórico de avarias desse equipamento. A plataforma SAP é usada como ferramenta de gestão de recursos da The Navigator Company, desde estruturação, bens materiais, compras, recursos humanos, notas de avaria, estatísticas, etc. A base de dados elaborada deve ser tratada e submetida a algoritmos de *machine learning* com visualização de resultados numa plataforma web que seja "*user-friendly*". Com os algoritmos de *machine learning* pretende-se relacionar a vibração de rolamentos, e a respetiva temperatura, com as frequências indicadoras de falha, previamente calculadas. E desta forma, prevê-se o tempo de vida de um dado equipamento (onde os rolamentos são o componente de maior taxa de falhas), permitindo as condições para a prevenir ou agendar a compra de peças sobressalentes atempadamente, reduzindo custos na aquisição de material/equipamento com entregas imediatas, isto é, de valor acrescido. Adicionalmente, existe uma vertente do projeto dedicado ao desenvolvimento de hardware, nomeadamente o sensor de vibração e temperatura, para verificar a viabilidade de um sensor de baixo custo.

1.4 Organização da dissertação

A dissertação é essencialmente dividida em 5 capítulos, sendo que no primeiro e presente capítulo é feita uma introdução e contextualização dos problemas a resolver. No segundo capítulo é apresentada a empresa, Complexo Industrial de Cacia da The Navigator Company, e exposta toda a informação que descreve a necessidade e a aplicabilidade de um sistema de manutenção preditiva. O terceiro capítulo é o estado da arte, onde são abordados, conceitos, tecnologias, e soluções atuais de manutenção preditiva. No quarto capítulo é apresentada a solução obtida bem como os respetivos testes práticos e resultados. Para finalizar encontra-se o quinto capítulo dedicado às conclusões e melhorias propostas.

2 A Empresa

2.1 História da Empresa

O Complexo Industrial de Cacia, da The Navigator Company, está localizado na vila de Cacia, distrito de Aveiro, no centro da maior mancha florestal de eucalipto em Portugal. Foi neste complexo que, em 1957, se produziu, pela primeira vez a nível mundial, pasta de papel a partir de eucalipto pelo processo Kraft, considerada excelente para o fabrico de papel de alta qualidade[1].

Figura 2-1 - Navigator Pulp Cacia[1]

A partir deste complexo industrial que na altura se chamava Companhia Portuguesa de Celulose se originou o grupo de 4 centros fabris hoje existente, Cacia, Figueira da Foz, Setúbal e Vila Velha de Ródão.

A sua produção atinge um volume anual na ordem das 350 mil toneladas de pasta branqueada de eucalipto, direcionada para a transformação em papéis especiais como *décor*, filtros, cigarros e *tissues* de alta qualidade. Integra também uma central de cogeração a biomassa associada à fábrica de pasta e uma central termoelétrica de biomassa para a produção de energia renovável. A The Navigator Company é internacionalmente reconhecida pelas suas pastas "desenhadas" para aplicações especiais, muito apreciadas pelos exigentes clientes europeus.

As máquinas principais que compõem a fábrica sofreram ao longo de décadas vários *upgrades* e o processo em si também tem sofrido melhorias. Não descartando a melhoria contínua, mas na mesma visão de aumentar a produtividade com cada vez menos matéria prima (reagentes, água, energia, etc.), existe a necessidade reduzir o tempo máximo de paragens de produção devido a avarias, por isso estamos perante uma clara necessidade da aplicação dos conceitos da Industria 4.0 tal como poderemos ver mais concretamente na secção seguinte.

2.2 Aplicabilidade e Processo

A figura seguinte (Figura 2-2) esquematiza de uma forma simplificada o processo de formação de pasta de papel, seguida de uma breve descrição dos passos que a constitui.

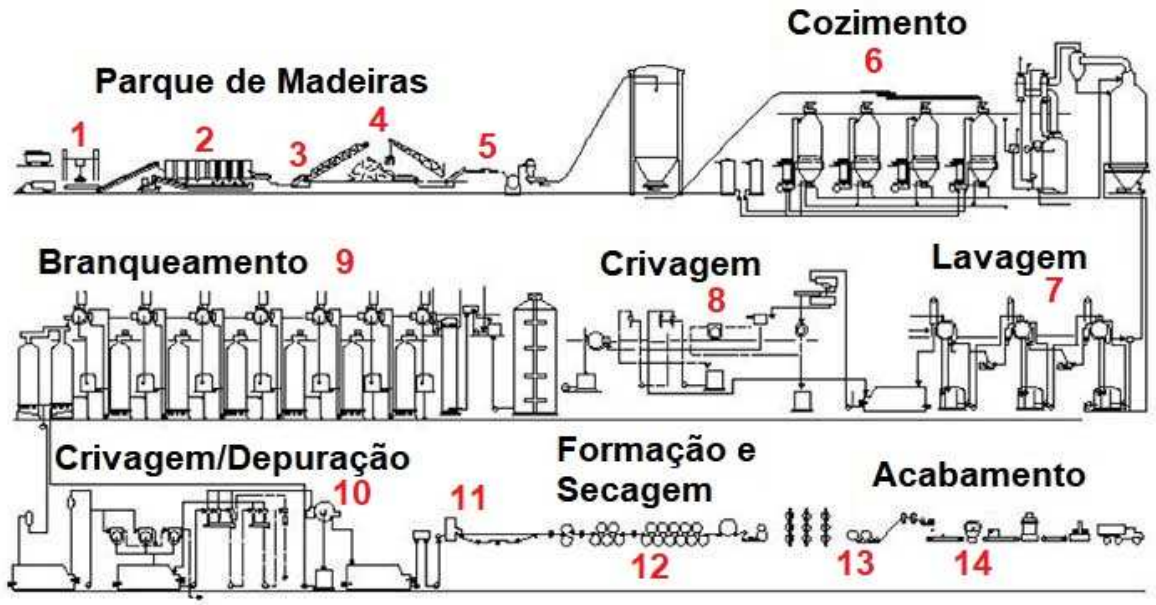


Figura 2-2 - Processo de fabrico de Papel [2]

O processo de transformação desde os troncos, ou também denominados toros, de eucalipto até à pasta de papel em estado solido para venda como matéria prima segue a sequência numerada de 1 a 14 na Figura 2-2, legendada pelos parágrafos seguintes.

1. Receção de matéria prima, toros de eucalipto.

   Aqui é recebida a matéria prima e aferida a sua densidade com recurso a uma grua fixa e um tanque de água, aplicando a lei da impulsão, pesando a madeira, fora e submersa em água.
2. Descasque de toros/troncos.

   Este passo não é necessariamente obrigatório porque existem duas linhas de produção de estilha de madeira, uma que recebe madeira descascada e, portanto, não possui uma máquina para este efeito, linha nº1 e a linha nº2 onde acontece o contrário.
3. Destroçamento.

   Nesta fase, os toros depois de estarem sem casca, e lavados, entram num destroçador, formando pequenas aparas de madeira com o tamanho máximo de 4 a 5 cm.

Um pequeno facto, o destroçador da linha nº2 é a máquina que exige mais binário na fábrica, sendo movido por dois motores de 800KW embraiados na mesma caixa redutora.

4. Empilhamento das aparas.

   Nesta fase as aparas são amontoadas ou "empilhadas", com máquinas do tipo *wheel loaders* de porte médio, por forma a garantir que existe sempre material disponível para a máquina do passo seguinte.

5. Captação de aparas.

   A captação de aparas é feita com um conjunto de "pás" (captadores) acionadas hidraulicamente e que se encontram subterradas pelos montes de aparas. O movimento alternado e cíclico dos captadores garante uma alimentação uniforme ao tapete de aparas que vai para a zona de cozimento.

6. Cozimento de aparas.

   O papel é constituído por fibras de madeira que só se conseguem obter pelo processo de cozimento desta com lixívia branca, um reagente químico constituído essencialmente por hidróxido de sódio (NaOH) e sulfureto de sódio (Na_2S).

   Este cozimento pode ser contínuo ou descontínuo e tem-se como produtos resultantes, lixívia negra e pasta crua. Parece um pouco contraditório, mas após o cozimento a pasta é denominada de pasta crua por ainda não ter sido submetida ao processo de branqueio e não de cozimento como o nome sugere. A lixívia resultante é negra, porque adquiriu essa cor ao reagir com a madeira ficando repleta de partículas orgânicas não relevantes para o processo de produção de pasta. Contudo existe um processo em paralelo que é a concentração dessa lixívia, por forma a ganhar um teor de sólidos suficiente para queima numa caldeira de recuperação. A mistura de produtos químicos em fusão resultante da queima, conhecida por *smelt*, passa a ser denominada de lixívia verde pela cor que apresenta nesta fase do ciclo de recuperação. Esta é enviada para a área de caustificação, onde é transformada novamente em lixívia branca.

7. Pré-Lavagem para retirar a lixivia em excesso.

   Embora não esteja presente na figura, a pasta após o cozimento, é crivada para remoção das frações de incozidos (nós) e aglomerados de fibras. Só depois é que esta passa pela pré-lavagem em contracorrente.

8. Crivagem.

   Aqui a pasta passa por 2 a 3 estágios de crivagem para retirar nós e outro tipo de sólidos que não são fibras e é então depois prensada por forma a retirar condensado/lixívia fraca em excesso, mantendo uma concentração baixa porque o meio de transporte da pasta é a água.

Isto é, quando é transmitida de uma zona para outra, como da zona de cozimento para o branqueamento, vai suspensa em água.

9. Branqueamento da pasta crua.

   Nesta etapa do processo, a pasta é branqueada por ação de agentes oxidantes que reagem com a pasta, fazendo-se alternar os estágios de reação química com estágios extração alcalina.

10. Crivagem e depuração.

    A crivagem e depuração da pasta, têm como função remover as pequenas impurezas que, eventualmente, ainda acompanhem a pasta.

11. Zona húmida da formação de pasta de papel.

    Esta zona é constituída por um formador de forma redonda (cilindro de sucção), seguido de três prensas, sendo que apenas a terceira, é revestida com feltros superior e inferior.

12. Zona seca da formação de pasta de papel.

    Esta zona da máquina é constituída por um secador e cilindros, aquecidos interiormente por vapor. Na entrada do secador a folha de pasta tem uma secura que ronda os 52 a 53% sendo o objetivo da saída do mesmo, 90%.

13. Corte e empilhamento de folhas de pasta de papel.

    Na saída do secador, a folha de papel é cortada em tiras longitudinais seguidas de quatro cortes transversais formando quadrados de aproximadamente 1 m^2. Estes são empilhados e enviados para zona de acabamento.

14. Zona de acabamento

    Cada pilha de folhas de pasta de papel, é pesada (aproximadamente 250 Kg), prensada, embalada e aramada, formando um fardo de pasta de papel. Posteriormente são empilhados 4 fardos e aramados em grupos de 8 fardos novamente formando uma *Unit*.

No passo 11 da produção de pasta de papel (Figura 2-2) que corresponde à Zona Húmida, os equipamentos têm sofrido um desgaste anormal face aos outros equipamentos da máquina de papel. Existem diversos tipos de rolos na máquina e com funções distintas, como por exemplo, os rolos esticadores, rolos guia, rolos corretores etc. representados pela área delimitada a vermelho na Figura 2-3. Nessa área, delimitada a vermelho, existem ao todo 14 rolos, isto é, 28 chumaceiras sendo que algumas têm mais do que 1 rolamento.

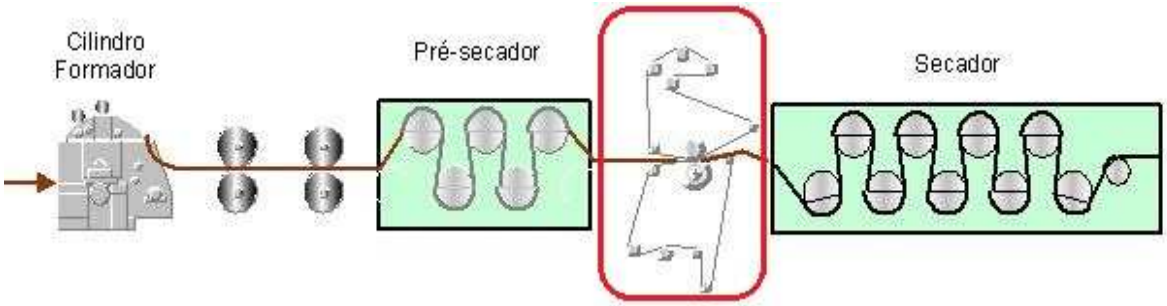


Figura 2-3 - Esquema simplificado da zona húmida e secagem da Tiragem 4

Segundo o histórico, até à data, de notas de avaria os rolos esticadores, são os que têm apresentado mais problemas nos rolamentos. Este tipo de rolos tem a função de esticar os feltros, superior e inferior, que guiam a pasta de papel uniformemente. Um exemplo de um rolo esticador é o da seguinte Figura 2-4.

Figura 2-4 - Chumaceira do rolo 4PR24 (Rolo esticador)

Como esta máquina funciona a velocidades reduzidas o estudo da espectrometria das vibrações com Transformadas de Fourier pode não ser possível distinguir as frequências características de uma avaria do ruido de fundo. Contudo isto não implica que o sistema desenvolvido não possa ser usado para monitorizar e prever avarias em outras máquinas de elevada criticidade para o processo, cujas velocidades são maiores. Tais como o ventilador de ar secundário da Caldeira de Recuperação

(mencionado na secção 1.3) ou as várias bombas de vácuo, que por exemplo, são necessárias para o funcionamento do Cilindro Formador que se pode ver na figura Figura 2-3 (à esquerda). Este é um cilindro aspirante e por tanto necessita de vácuo. Sem vácuo, fica essa linha de produção de pasta de papel parada.

Para prever a avaria dos rolamentos que constituintes desta máquina, é necessário o estudo de vibrações no domínio da frequência, em especial para baixas frequências. Com recurso à amostragem de temperatura e aceleração em tempo real, e aplicação de um algoritmo preditivo pretende-se obter uma relação entre a temperatura, vibração no domínio da frequência e as frequências de falha dos rolamentos (detalhadas na secção 3.4). Adicionalmente, pretende-se relacionar o histórico de avarias num dado equipamento com a base de dados criada. A monitorização da vibração dos rolamentos em tempo real como forma de determinar o estado de uma máquina e prever uma avaria, pode ser aplicado a qualquer máquina que tenha um elevado nível de criticidade e que acarrete um elevado custo de manutenção e perda de produção numa paragem não planeada.

3 Estado da arte

3.1 Manutenção na Industria 4.0

O termo Indústria 4.0 surgiu, como designação para o passo seguinte na evolução da indústria, um conjunto de recomendações apresentado pela primeira vez na Hannover Messe[1], Alemanha. Nestas recomendações são salientadas a integração de sistemas, a *Big Data* e a Internet como meio de interligação entre equipamentos na industria. A progressão e principais marcos são apresentados na Figura 3-1.

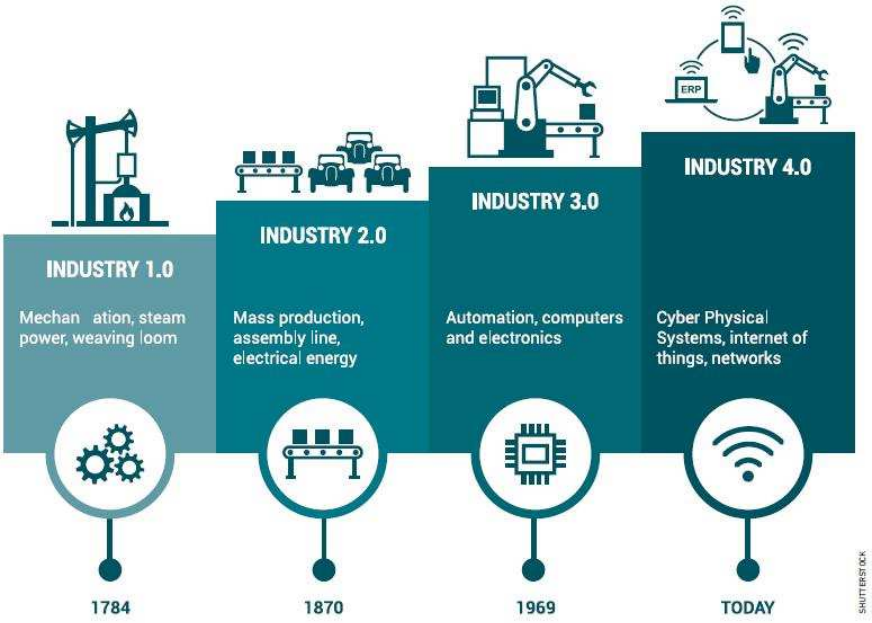


Figura 3-1 - Evolução da indústria [3]

Na Figura 3-1, os marcos que distinguem as diferentes Eras da indústria são, os engenhos mecânicos, principalmente movidos a vapor (Industria 1.0), a produção em série (Industria 2.0), o transístor e todos os avanços tecnológicos a este associados (Industria 3.0) e a interligação como meio de partilha ou transmissão de informação em ambiente industrial (Industria 4.0).

[1] Hannover Messe é das maiores feiras de tecnologia industrial no mundo.

Além da internet e da partilha de informação a um nível que abrange os mais simples dispositivos, existem outros conceitos que em conjunto formam os 9 pilares da Industria 4.0. Dos 9 pilares, na Figura 3-2, apenas 2 são abordados neste projeto a *Big Data* e *Internet of Things* (IoT) ou *Industrial Internet of Things* (IIoT), no caso da industria. Estes dois conceitos são os que mais contribuem de forma direta para a manutenção preditiva, sendo que os outros pilares, como *Simulation, Autonomous Robots* e *Augmented Reality*, contribuem de forma direta para o aumento de produção.

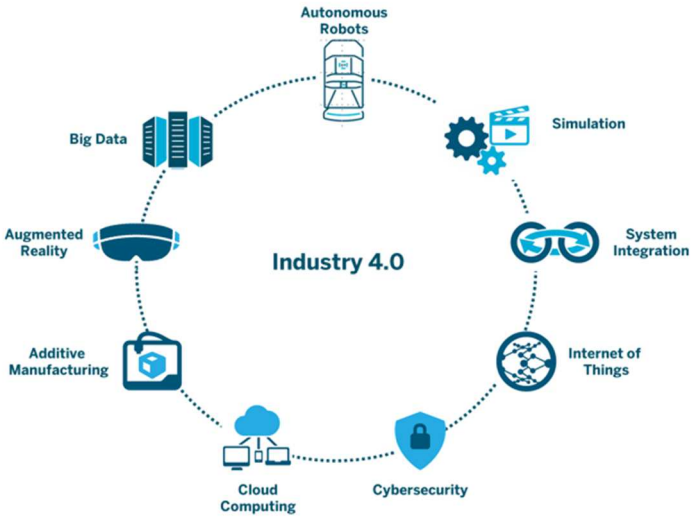


Figura 3-2 - Os 9 pilares da indústria 4.0

3.1.1 A internet das coisas na Indústria (IIoT)

O conceito *Internet of Things* (internet das coisas) surgiu com a necessidade que a sociedade tem de estar dependente da ligação à internet. Se todos estiverem ligados à internet a ligação entre dispositivos, que permita melhorar o quotidiano, é facilitada. Com um *smartphone*, hoje em dia podemos abrir a porta da garagem, regular a iluminação de casa, monitorar o sistema de rega do jardim, etc. Tudo isto porque estão ligados na mesma rede. Rapidamente a Industria olhou para este conceito com outros olhos, visto que se conseguirmos que todos os equipamentos comuniquem entre sim, conseguimos atingir um nível de produtividade e de manutenção que de outra forma não seria alcançável. Por isso é que a então designada IoT, ou no caso da industria IIoT, é um dos pilares da Industria 4.0, como se pode verificar na Figura 3-2.

3.1.2 Big Data

As bases de dados são outro pilar da Industria 4.0, como se verifica na Figura 3-2, mais propriamente o conceito de "*Big Data*" que significa, o armazenar toda a informação fisicamente ou via *Cloud*. A *Big Data* não só acelera processos morosos que foram informatizados, mas também serve de ferramenta para determinação estatística de diversos indicadores numa indústria. Estes indicadores podem ser a produtividade de uma máquina, o seu tempo de vida, qual o principal fator responsável por uma avaria recorrente, etc.

Uma base de dados é uma coleção de dados armazenados num servidor. Estas, quando têm uma elevada complexidade de relação, por forma a não sobrecarregar o sistema com informação redundante, são desenvolvidas utilizando técnicas de modelação tal como a normalização.

Para que um utilizador interaja com a base de dados no sentido de a modificar, tem de recorrer a um gestor de base de dados, também denominado, do inglês pela sigla DBMS, *database management system* como por exemplo o phpMyAdmin [4].

A maioria das aplicações de bases de dados responde a uma linguagem por *queries*. Cada sistema de base de dados compreende a sua linguagem *query* e converte cada *query* para um formato exequível pelo servidor por forma a obter resultados (resposta da *query*).

Ao conjunto do servidor e do respetivo DBMS podemos chamar de Sistema de base de dados. Alguns dos sistemas de base de dados mais conhecidos são, Oracle, DB2, Microsoft SQL Server, MySQL e Ingres. Estando estes dois últimos ao abrigo da GNU General Public Licence sendo, portanto de acesso livre.

A escolha do servidor MySQL deve-se ao facto de este ser gratuito e também pelo seu reconhecimento. O MySQL é utilizado em 9 dos 10 sites mais acedidos mundialmente tais como o Facebook, Twitter, Wikipedia, Youtube. É incluído no pacote LAMP (para Linux) e XAMP (para Windows) sendo este escolhido por lideres de IT experientes, como forma de melhorar a eficiência operacional e reduzir custos, como se pode ver na comparação entre o custo total durante 3 anos de um sistema MySQL ou Microsoft SQL Server, na figura seguinte.

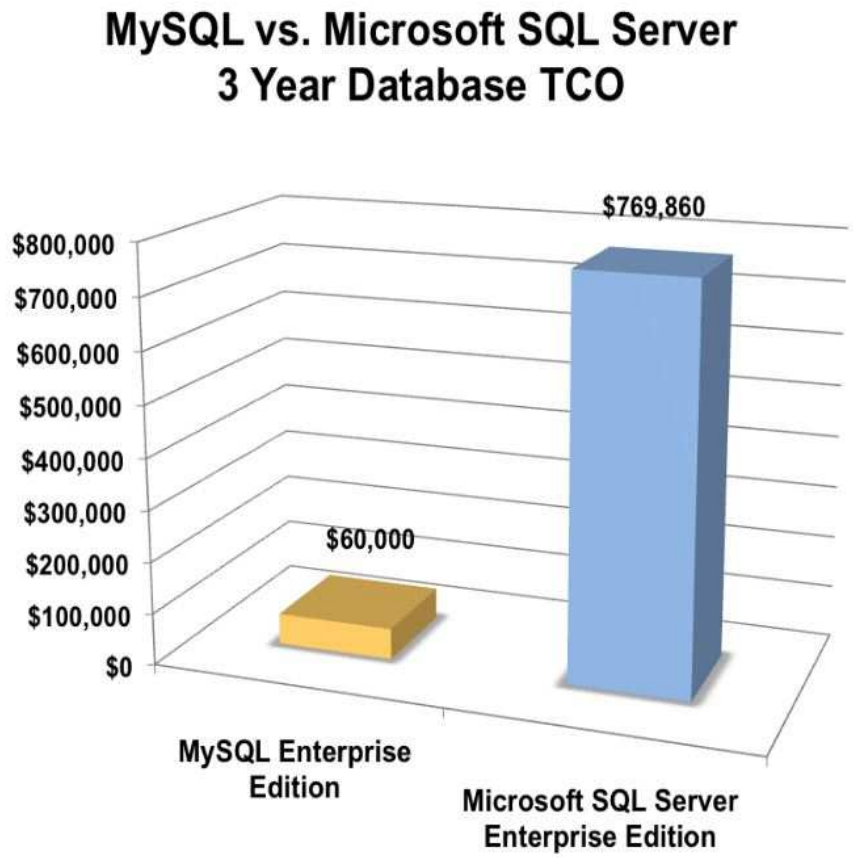


Figura 3-3 - MySQL vs. Microsoft SQL Server[5]

3.1.3 Tipos de Manutenção

De acordo com a norma europeia European Standard EN 13306 [6], a manutenção é dividida em dois grandes tipos:

- Manutenção corretiva: este tipo de manutenção é aplicado após a deteção de falha. Uma fábrica, que apenas adote este método de manutenção, não pode ser melhorada, contudo as falhas podem ser corrigidas com alguma rapidez, associada ao conhecimento (*know how*) que os técnicos têm visto que o número de falhas é máximo quando comparado com os outros métodos de manutenção. Com uma elevada cadencia de avarias e uma necessidade de as corrigir cada vez mais rapidamente, os técnicos de manutenção otimizam a forma como intervêm com cada intervenção, isto é, com conhecimento adquirido. Quando um determinado equipamento

tem performance decrescente no desempenhar da manutenção corretiva, então este seguirá o efeito de bola de neve, conduzindo a custos agravados.[6]

- Manutenção preventiva: este tipo de manutenção aplica-se antes da deteção de falha. Consiste em atividades de manutenção repetidas com um certo intervalo de tempo, baseadas num plano de manutenção. O plano de manutenção (rotinas) é elaborado maioritariamente consoante o número de horas de trabalho de um dado equipamento, e pode ser modificado com base em outros indicadores que encurtam o tempo de vida como por exemplo a medição (descontínua) de vibração, pressões e temperaturas. com técnicos qualificados para fazer o estudo e fazer uma classificação.

A Manutenção preventiva, acima mencionada, subdivide-se em vertentes tais como a Manutenção Sistemática, Planeada, Preditiva, entre outros. Destas vertentes, a Manutenção Preditiva é a metodologia de manutenção que se pretende estudar. Esta consiste num acompanhamento do equipamento de forma continuada monitorizando indicadores que reflitam o estado atual desse equipamento. Nesta existe também um algoritmo de classificação ou regressão que correlaciona os indicadores por determinando o tempo útil de uma máquina. Os principais objetivos da manutenção preditiva são:

- Determinação antecipada da necessidade de manutenção;
- Eliminar desmontagens desnecessárias para inspeções periódicas;
- Aumento do tempo de disponibilidade dos equipamentos;
- Redução de trabalho urgente não planeado;
- Aproveitamento da vida útil total de um equipamento;
- Determinação prévia das interrupções de produção para manutenção.

Na era de máquinas industriais 4.0, os equipamentos de controlo e monitorização detetam automaticamente uma falha numa fase inicial em alguns dos seus mecanismos, mesmo que ainda não seja subtil e não afete o desempenho da máquina. Se uma falha potencial for detetada – como uma vibração fraca num rolamento – o software destas máquinas 4.0 calcula as tendências e as repetições desse problema, e prevê a data futura em que esta vai afetar a precisão ou desempenho da máquina. Comparando os tipos e manutenção com o custo para reparar uma determinada avaria, obtém-se o gráfico seguinte da Figura 3-4.

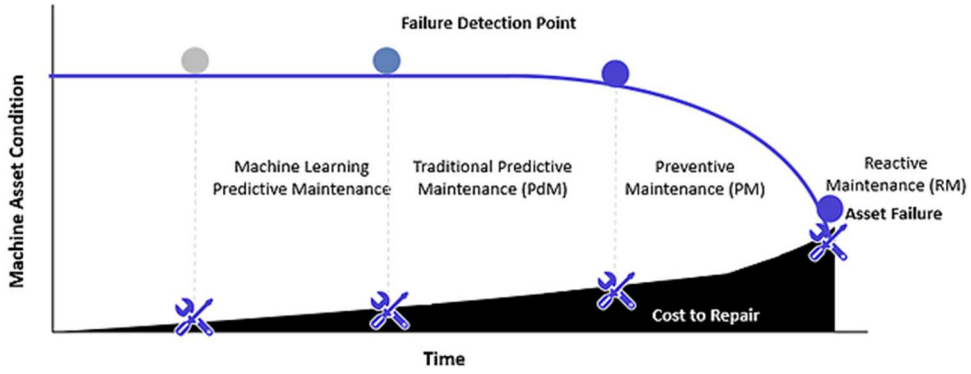


Figura 3-4 - Gráfico temporal do custo para repara uma avaria vs. tipos de manutenção [7]

É importante referir que deve ser encontrado um equilíbrio entre o investimento na industria 4.0 com base na criticidade ou importância que um dado equipamento apresenta para um determinado processo. Investir deliberadamente, pode conduzir a uma redução abismal de manutenção reativa, mas pode também resultar num retorno do investimento bastante demorado.

3.2 Algoritmos preditivos

Um algoritmo não é nada mais do que uma sucessão de condições lógicas baseada em operações matemáticas e/ou lógicas que chegam a um determinado fim. Como podemos ver no exemplo da Figura 3-5.

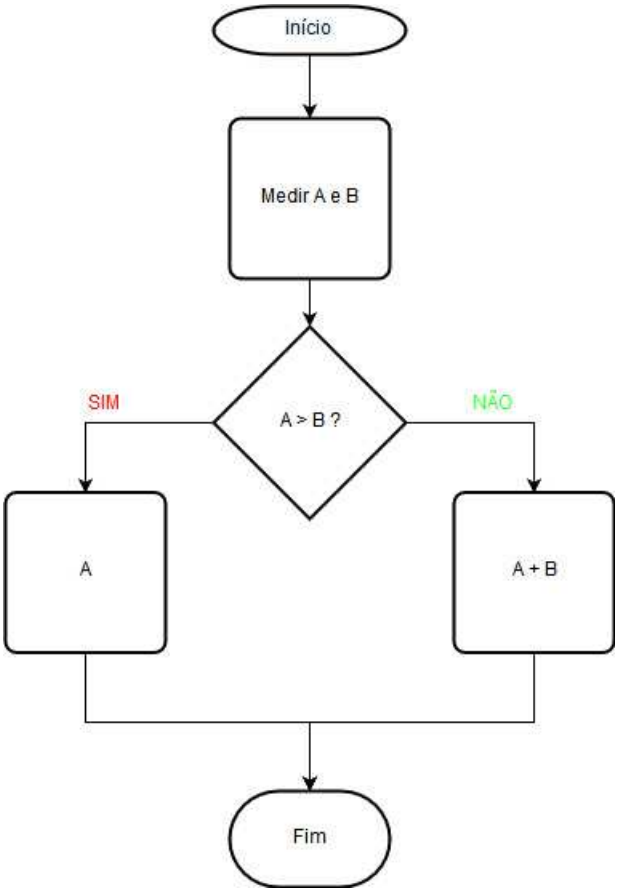


Figura 3-5 - Algoritmo Genérico

No fluxograma da Figura 3-5 verificam-se operações encadeadas de forma a chegar a determinado valor.

No âmbito da manutenção preventiva, a figura seguinte (Figura 3-6) exemplifica como é que este tipo de manutenção se processa atendendo ao algoritmo apresentado.

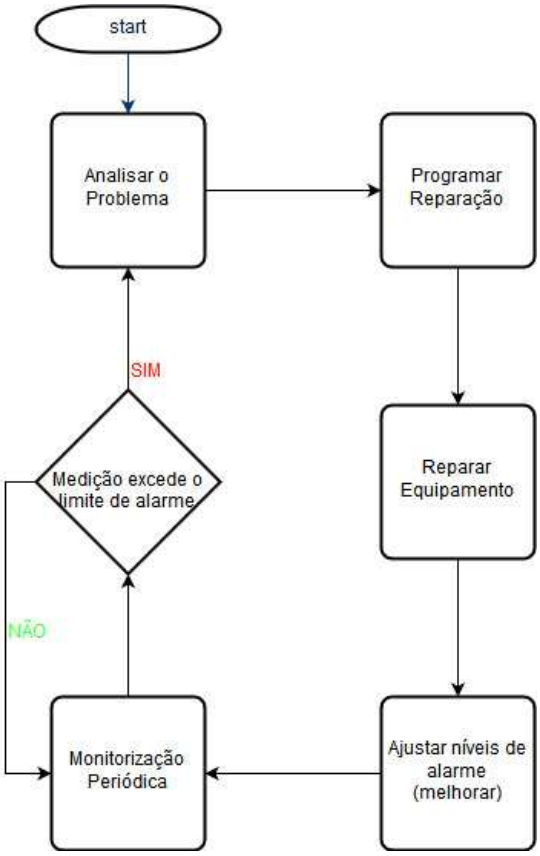


Figura 3-6 - Algoritmo básico de manutenção

Na Figura 3-6 começando na monitorização periódica, assume-se aqui, que uma ou várias grandezas físicas são monitorizadas (temperatura, pressão, tempo, deslocamento, etc.). Segundo um tempo pré-definido de ciclo, verifica-se se os valores medidos ultrapassam um limite estipulado. Se não ultrapassar, continua a monitorar, se sim, então é agendada uma reparação e é ajustado o alarme mediante o tempo de trabalho entre a última avaria e a atual, denominada de *Mean Time Between Failures* (MTBF).

Este algoritmo não verifica a tendência dos valores medidos, está confinado a um ou vários limites onde pode resultar num alarme. E tal como se verifica na Figura 3-7, a variável passa do limite superior duas vezes sendo que estes podem ser falsos alarmes atendendo à tendência dos valores medidos. Isto implica a análise por um especialista, enquanto que a manutenção preditiva analisa em tempo real, verifica a cadência de valores, as diferenças etc., tudo isto com o *Machine Learning.*

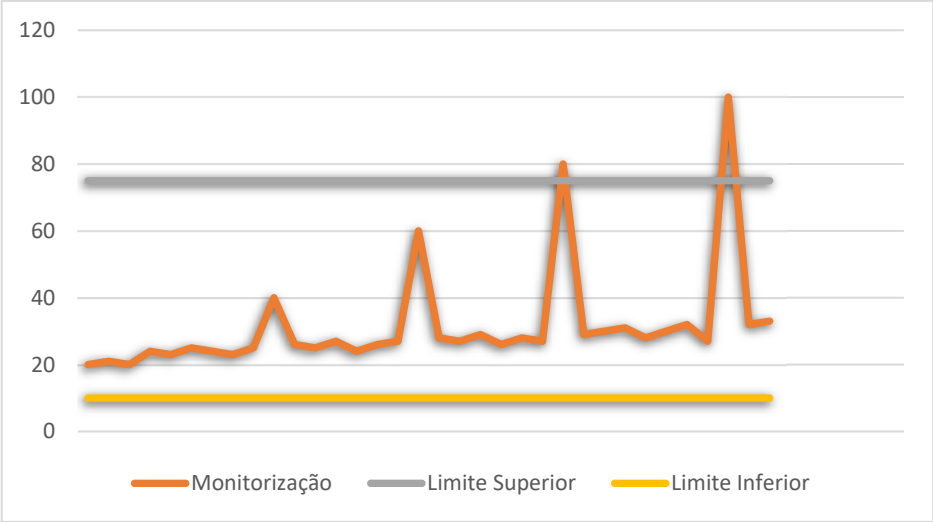


Figura 3-7 – Análise de tendência vs. análise de limites

3.3 Machine Learning

O *Machine Learning*, isto é, a "aprendizagem da máquina". é uma categoria de algoritmo que extrai padrões de uma base de dados com base em cálculos estatísticos.

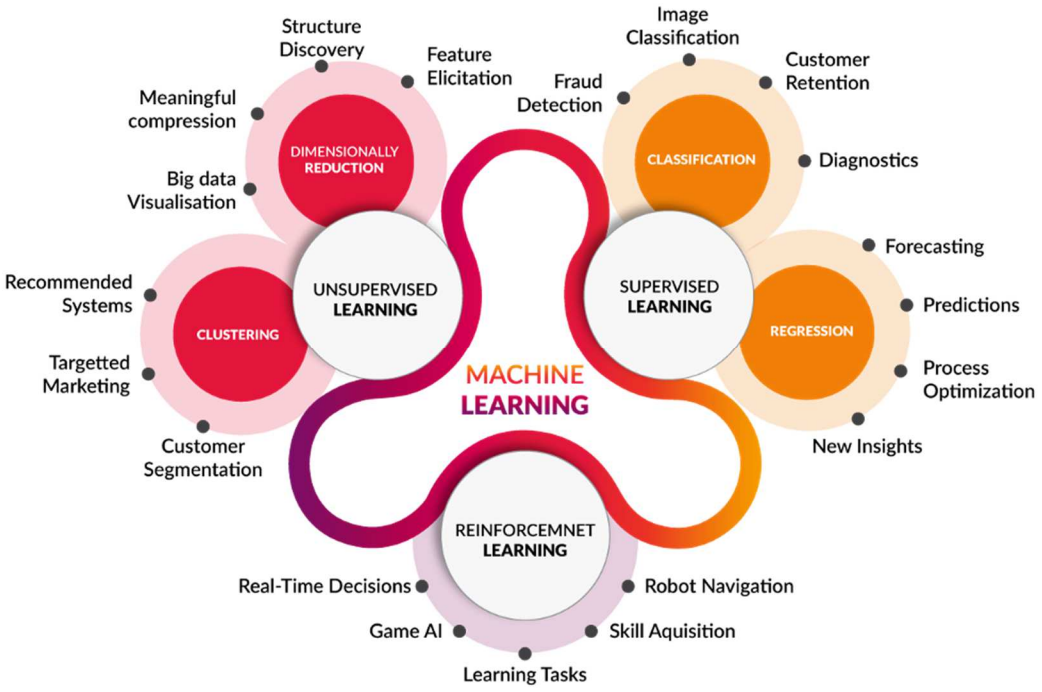


Figura 3-8 - Categorias de machine learning e suas aplicações [8]

A Figura 3-8 apresenta o tipo de problemas que podem ser solucionados através de métodos de *machine learning*, dispostos na orla da figura, tais como *Learning Tasks, Predictions,Targetted Marketing*, etc. Estes problemas ou aplicações estão associados a uma divisão do *machine learning* em 3 grandes categorias, a aprendizagem supervisionada (*supervised learning*), aprendizagem não supervisionada (*unsupervised learnig*) e aprendizagem reforçada (*reinforcement learning*). A aprendizagem supervisionada é a aproximação de dados a uma função, como por exemplo uma regressão linear, onde a partir de vários pontos são relacionadas, de forma linear, as diferentes variáveis. Nesta, são fornecidos dados catalogados, isto é, dados de entrada em que se sabe o resultado num algoritmo preditivo. Estes dados são na maioria das vezes uma tabela de treino (*training set*) em que se separa a coluna resultados (*labels*), submetendo os dados a um algoritmo que de alguma forma aproxime o resultado à coluna separada inicialmente.

Aprendizagem não supervisionada tem como objetivo determinar o que existe de comum entre determinados pontos/dados de modo a catalogá-los por grupos ou classes. Segundo Tseng [9], um exemplo é um algoritmo de aglomeração (*cluster algorithm*) onde os dados são apresentados sem se saberem quaisquer tipo de relações entre os mesmos, sendo posteriormente encontradas relações que definem uma estrutura dando origem a grupos. Um fator determinante neste tipo de algoritmos é a probabilidade da distribuição.

A aprendizagem reforçada tem como objetivo aprender mediante um cálculo de uma pontuação final tendo em conta penalizações ou bonificações resultantes das ações tomadas, e é por este motivo que vulgarmente é explicada através de um jogo cujo o computador tem de aprender a jogar sem sequer saber as regras tendo como objetivo a melhor pontuação possível[10]. A aprendizagem reforçada entra já no âmbito da inteligência artificial.

Como se pode ver também na Figura 3-8, sabendo que o âmbito deste projeto é a previsão de uma avaria, identificamo-nos com a aprendizagem supervisionada (*Supervised Learning*), onde temos os termos no ramo de regressão, Prognosticar (*Forecasting),* Previsão (*Predictions)* e Otimização de Processo *(Process Optimization)* e no ramo da classificação, Diagnósticos (*Diagnostics)* e Deteção de Fraude (*Fraud Detection*).

Tendo como enfoque a aprendizagem supervisionada alguns dos algoritmos, abordados nas secções seguintes, são:

- Naive Bayes;
- Decision Trees / Random Forests;
- Hiden Markov Models (HMM);
- Suport Vector Machines (SVM);
- Neural Networks.

3.3.1 Naive Bayesian

O classificador de Naive Bayesian segue o teorema de Bayes também conhecido por probabilidade condicional invertida. Por outras palavras, Bayes formulou uma maneira de diminuir uma dada população a analisar mediante o número de informações que se tem sobre essa população e de que forma se podem traduzir em cadeias probabilísticas (probabilidade de acontecimentos em cadeia)[11].
Fundamentalmente, uma inferência Bayesiana é descrita por[9]:

$$P(\theta|X) \tag{3.1}$$

Onde os parametros θ são desconhecidos, sendo por isso expressos como uma distribuição de probabilidade, face às observações X. Esta probabilidade é conhecida por probabilidade a posteriori ou distribuição de probabilidade posterior.
Segundo Chris Fonnesbeck [9], devem ser especificadas as distribuições de probabilidade para uma dada amostra e para os parâmetros desconhecidos que a caracterizam. Este processo envolve várias suposições. De seguida é calculada uma distribuição posterior, que muitas vezes não pode ser calculada analiticamente, recorrendo a outros métodos como por exemplo o da simulação. A partir da distribuição posterior, é possível calcular estimativas pontuais, intervalos de confiança e até fazer previsões. Por fim, por causa de terem sido feitas suposições nos parâmetros iniciais, deve-se testar o modelo para verificar se este se ajusta aos dados de forma razoável. Na estatística Bayesiana, o único estimador é a formula de Bayes, também conhecida por Regra de Bayes ou Teorema de Bayes definida pela expressão seguinte (Eliezer S. Yudkowsky [12]):

$$P(H|D) = \frac{P(H) \times P(D|H)}{P(D)} \tag{3.2}$$

Onde:

- H é a hipótese/parâmetros, vulgarmente representada por θ;
- D são os dados a analisar;
- P(H) = probabilidade das hipóteses antes das observações feitas aos dados (*priori*);
- P(H|D) = probabilidade das hipóteses depois das observações feitas aos dados (*posteriori*);

- P(D|H) = probabilidade dos dados segundo uma dada hipótese.
- P(D) = probabilidade dos dados segundo qualquer hipótese Constante de Normalização.

3.3.2 Decision Trees e Random Forests

Decision Trees são tal como próprio nome indica, tomadas de decisão ramificadas. Estas são escalonadas com base num conhecimento prévio sobre uma dada população ou informação que se pretende classificar. Essas tomadas de decisão, tendem a ramificar ganhando a forma de uma árvore invertida, como indica a Figura 3-9. A forma como se podem dividir os dados para cada tomada de decisão, pode ser feita recorrendo a outros algoritmos como “greedy algorithm” ou até o índice de Gini.

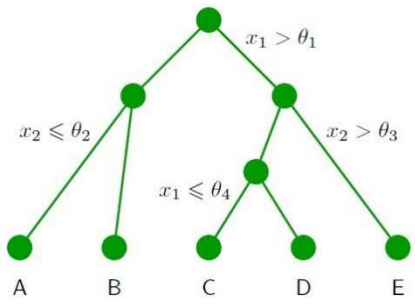


Figura 3-9 - Árvore binária [10]

Também é possível construir várias árvores de decisão sobre uma determinada base de dados sendo que o resultado de uma previsão é feito por votação, isto é, são percorridas todas as árvores e o resultado que ganhar a maioria é o resultado final. Isto acontece nas chamadas Bagged Decision Trees (*bagging* = *bootstrap-aggregation*).

3.3.3 Hidden Markov Chains

Segundo C. Bishop [10], os modelos Hidden Markov são algoritmos baseados no método de Markov que se divide em 3 fases, avaliação, descodificação e aprendizagem. Este tipo de modelo é

frequentemente usado em simulações porque contrariamente às redes neuronais, não se torna extremamente complexo a cada nó acrescentado, visto que é baseado em matrizes de probabilidade (computacionalmente, os cálculos matriciais são mais rápidos). Outra particularidade é de não necessitar de um conhecimento alargado sobre a base de dados tal como no modelo de Bayes ou Decision Trees. Este modelo tem como base na sua formulação a sucessão de probabilidades condicionadas expressa em (3.3), neste caso, de primeira ordem.

$$p\left(z^{(m+1)}\middle|z^{(1)},\dots,z^{(m)}\right)=p\left(z^{(m+1)}|z^{(m)}\right) \tag{3.3}$$

Onde Z é uma dada observação e $p(Z^{(m+1)}|Z^{(m)})$ é a probabilidade dependente de $Z^{(m)}$[10].

3.3.4 Support Vector Machines

Support Vector Machines (SVM) é um algoritmo que tal como o Random Forests ou Decision Trees, quando usados como classificadores, baseia-se na logica de *threshold* (limites) que divide os dados em múltiplas partes ou classes. Contudo, no algoritmo de SVM não existe apenas um limite, mas sim uma margem delimitada por dois vectores (*support vectors*) como de pode ver na Figura 3-10, sendo estes vetores a mínima distância entre os dados de amostragem e o limite de decisão[10].

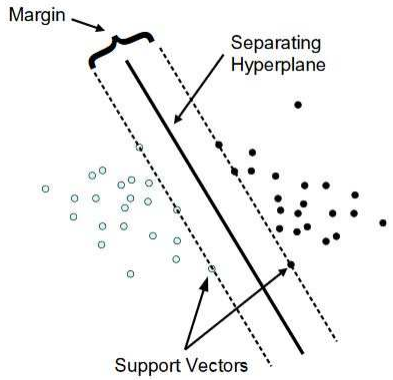


Figura 3-10 – Representação gráfica de SVM [13]

A ideia é de separar linearmente dados num espaço multidimensional com um hiperplano que quando projetados em duas dimensões seriam impossíveis de separar linearmente como se pode ver na Figura 3-11.

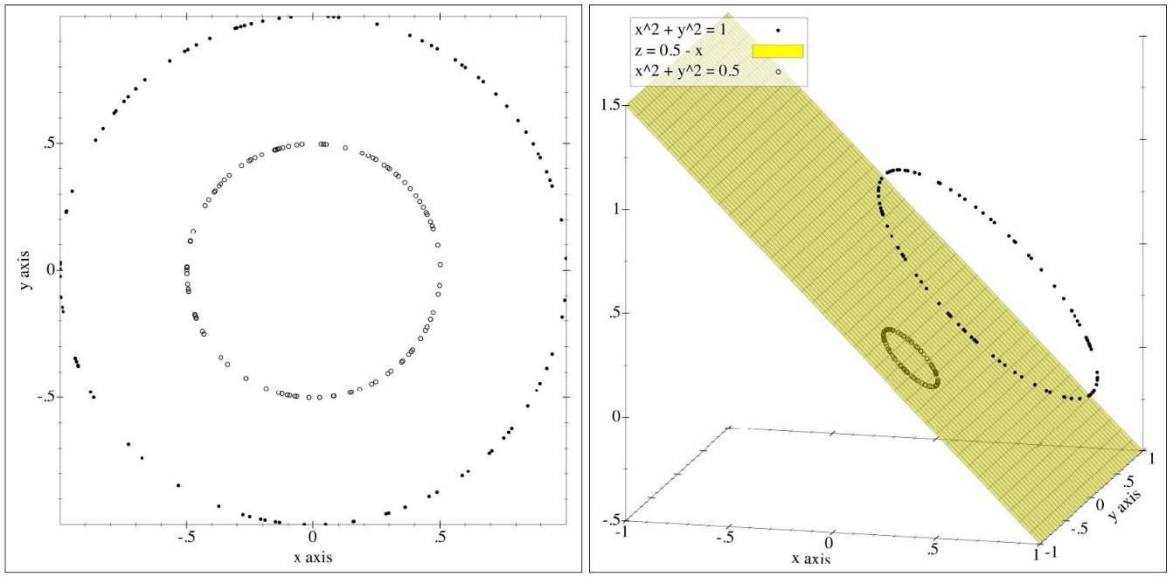


Figura 3-11 - Separação de dados por um plano [14]

3.3.5 Redes Neuronais Artificiais

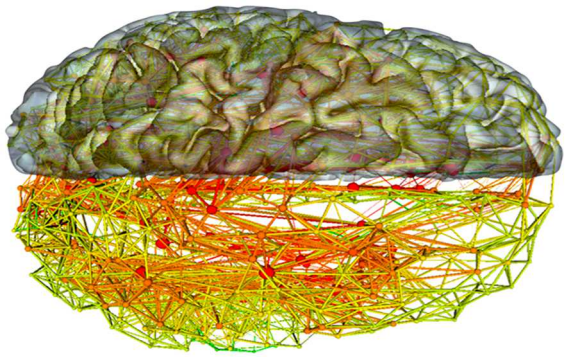


Figura 3-12 - Cérebro como inspiração para as redes neuronais artificiais [44]

As redes neuronais artificiais (RNA ou do inglês *Artifitial neural network* ANN) são conceptualmente baseadas a partir de uma rede neuronal biológica tal como o cérebro humano (Figura 3-12). As redes neuronais são compostas por neurónios que enviam sinais de uns para os outros como resposta a determinado estímulo (*input*).

Em 1943 Warren McCulloch e Walter Pits apresentaram o primeiro modelo matemático de um neurónio artificial e em 1958 Frank Rosenblatt descreveu pela primeira vez o conceito de um perceptrão (*perceptron*). Um perceptrão é uma unidade com entradas (*inputs*), ponderadas por pesos, que produz uma saída binária segundo um determinado limite. Um nó numa RNA ou um neurónio (artificial) é uma generalização do conceito de um

perceptrão. Este é também um somador de entradas em que cada uma é afetada pelo respetivo peso sináptico, mas ao invés de produzir um resultado binário, produz um valor entre 0 e 1 com base no quão aproximados os valores de entrada correspondem a uma determinada categoria ou padrão. Os nós são normalmente afetados por uma constante que lhe é aritmeticamente somada (*Bias*) por forma a favorecer a aproximação dos extremos da função de ativação (abordada nos seguintes parágrafos).

Tal como referido anteriormente, matematicamente um neurónio artificial é representado pela equação (3.4) [14]:

$$a = \sum_{i} \omega_i x_i + b \tag{3.4}$$

Onde a saída (a) depende do quanto cada atributo (valor de entrada, x_i) contrui, através do somatório dos atributos, multiplicados por um coeficiente denominado de peso ou também de peso sináptico, alusivamente ao neurónio biológico (ω), de seguida somados da constante *bias* (b).
Um nó apenas consegue representar funções linearmente separáveis. Verificou-se isto quando se revelou impossível fazer a aprendizagem do OU exclusivo (XOR) num único nó. Por isso sentiu-se a necessidade de criar camadas intermédias entre a saída e a entrada que processassem o output desejado com base na "informação transmitida" de um nó para outro.
Uma RNA definida na Figura 3-13, possui neurónios de entrada, neurónios intermédios e neurónios de saída, mas quando se trata de a classificar como rede de "N-camadas" apenas são incluídas as camadas intermédias e a de saída. Por outro lado, pode ser descrita pelo número total de nós de toda a rede. Desta forma podemos dizer que a figura seguinte representa uma rede neuronal de 2 camadas (2-*layer*) composta por (3,2,1) onde se constatam 3 neurónios de entrada, 2 neurónios na camada intermédia (ou escondida, do inglês *hidden layer*) e 1 neurónio na camada de saída. A esta rede também se dá o nome de perceptrão de multicamada (*Multi-layer Perceptron - MLP*).

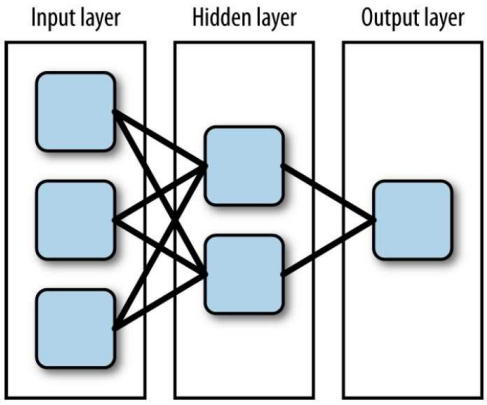


Figura 3-13 – Exemplo de uma RNA (3,2,1)[10]

Se à equação (3.4) for aplicada ao conceito da multicamada, então esta pode ser reescrita como:

$$a_{kj} = \sum_{i=1}^{N} \omega_{ji} x_i + b_k \tag{3.5}$$

Onde:

- k é o número da camada
- j é o número do neurónio
- i é o número dos neurónios anteriores

Outro aspeto topológico tem a ver com o sentido das ligações na rede, este diferencia uma rede de ser alimentada para a frente (do inglês *feed-forward neural network*) ou recorrente como é o exemplo da rede de Hopfield[15]. Neste trabalho apenas será usado o método *feed-forward*.
O resultado de cada neurónio é transformado usando uma função de ativação, como por exemplo a função sigmoid, tangente hiperbólica, Elliot, função gaussiana e ReLU (*Rectified Linear Unit*), dando origem à definição anterior de nó de uma RNA. As funções de ativação devem compreender valores entre 0 e 1, assimilando a probabilidade em que quanto mais próximo de 1, maior é a probabilidade de o próximo nó ficar ativo. A função sigmoid (3.6) é usada na área da aprendizagem de máquina (*machine learning*), e até em *deep learning* embora seja cada vez mais substituída por funções como a ReLU.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.6}$$

A função de ativação tem como objetivo verificar o quanto positiva é a soma ponderada, neste caso, a sigmoid tem uma forma em S e é diferenciável para qualquer número real. Adicionalmente, a sua derivada (3.7) é sempre positiva, como se pode constatar na Figura 3-14 designada por sigmoid prime.

$$\frac{d\sigma(x)}{d(x)} = \sigma(x) \cdot (1 - \sigma(x)) \tag{3.7}$$

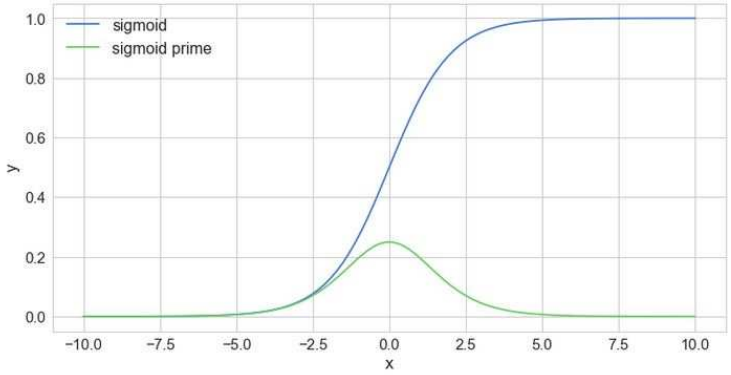


Figura 3-14 - Gráfico da função sigmoid (azul) e da sua derivada (verde)[16]

Aplicando a expressão(3.5) em (3.6), tendo em conta a rede definida na Figura 3-15, obtém-se a expressão (3.8) que neste caso representa uma rede neuronal de duas camadas, a de saída e uma camada escondida.

$$y_k(x, w) = \sigma\left(\sum_{j=1}^{M} \omega_{kj}^{(2)} \sigma\left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + x_0\right) + z_0\right) \tag{3.8}$$

Onde:

- M -> número de neurónios na camada escondida
- D -> número de neurónios na camada de entrada
- K -> número de neurónios na camada de saída
- j -> índice de um neurónio na camada escondida

- i -> índice de um neurónio na camada de entrada
- x_0 e z_0 -> *bias*
- k -> índice de um neurónio na camada de saída

Na Figura 3-15 verifica-se o sentido do fluxo de informação que passa da entrada para a saída da rede, marcado pelas setas verdes, mais uma vez a relembrar o conceito de alimentação para a frente.

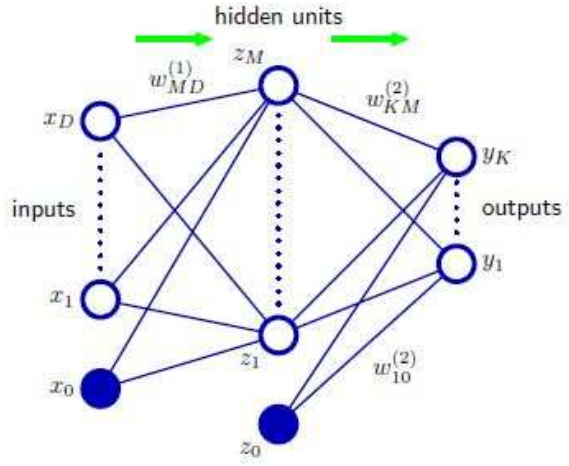


Figura 3-15 - Rede neuronal de 2-camadas [10]

Os *bias* são valores de *offset* (constante somada) que permite que uma ativação seja feita quando é superior a um determinado valor, de forma significativa. Segundo Donald Hebb [15], a ligação entre dois neurónios, isto é, cada peso sináptico, deve ser reforçada se esses dois neurónios estiverem ativos ao mesmo tempo. Outro parâmetro que se deve ter em conta é o ritmo de aprendizagem muitas vezes representado por (η). O ritmo de aprendizagem representa a velocidade de convergência dos valores dos pesos para a situação desejada. Se este for baixo, a convergência é lenta e se for elevado pode provocar oscilações indesejadas.

A atualização dos pesos da rede é feita através da minimização do erro, por uma função de custo sendo que as mais comuns são o erro quadrático (MSE) para a regressão, e entropia cruzada (*cross-entropy*) para a classificação. Os algoritmos baseados na descida do gradiente (regra Delta) são os mais usados no processo de minimização do erro quadrático. Um desses é o algoritmo de retro propagação do erro ou do inglês *back-propagation error*. A regra delta procura minimizar o erro quadrático, calculando a partir da diferença entre a saída desejada e a obtida, orientando as modificações dos pesos no sentido do gradiente descendente da superfície do erro. Sendo (a_j) o valor do nó da camada (j), (r_j) o resultado pretendido e (M) o número de neurónios, a equação (3.9) representa o erro (E) calculado entre uma camada intermédia e a saída[15].

$$E = \frac{1}{2}\sum_{j=1}^{M}\left(r_j - a_j\right)^2$$

(3.9)

O gradiente de do erro (∇E) é dado pelo vector formado a partir das derivadas parciais de (E) relativamente a cada um dos pesos. O gradiente dá a direção do maior aumento do erro e a direção oposta dará a orientação do decréscimo mais rápido do erro como se pode ver na equação seguinte onde mais uma vez aparece o fator de velocidade de aprendizagem (η).

$$\Delta w_{ij} = -\eta \frac{\delta E}{\delta w_{ij}}$$

(3.10)

Através da expressão (3.10) e (3.9) entende-se que o gradiente dependa da derivada parcial da função de ativação visto que (a), em (3.9), é o resultado da aplicação da função sigmoid. Por tanto o erro, como se pode ver no gráfico da derivada da função sigmoid (Figura 3-14), diminui á medida que nos aproximamos dos extremos. A diferenciação é necessária porque se pretende encontrar o mínimo local na função de custo. E por forma a atualizar os pesos tem-se que:

$$w_{ij}{}^{(t+1)} = w^{(t)} + \Delta w_{ij}$$

(3.11)

Onde (t) é número de épocas ou iterações que o algoritmo corre e (i) e (j) são os índices dos nós ligados. Quase como que um jogo jogado múltiplas vezes até se obter a pontuação perfeita, aprendendo em cada tentativa. A aprendizagem por retro propagação define-se então de uma forma generalizada pelo seguinte pseudocódigo:

1. Inicializar os pesos da rede
2. Definir uma condição de paragem (erro mínimo ou limite de iterações)
    2.1. Para cada par de treino, (dados de entrada, resultado espectado)
        2.1.1. Propagar os valores de entrada para a saída segundo a equação (3.5) submetida a função sigmoid.
        2.1.2. Calcular a variação dos erros por retro propagação, da camada de saída para a camada escondida e da camada de entrada para a camada escondida.

2.1.3. Atualizar os pesos

Existem ainda algumas formas de acelerar a convergência do algoritmo de retro propagação. Uma delas consiste na introdução de um termo determinado de forma heurística multiplicado pela variação do peso na expressão (3.11), designado de *momentum*, (μ)[15].

São vários os métodos que modificam o comportamento da aprendizagem, um já supracitado diz respeito à velocidade de aprendizagem e outro importante de referenciar é o método Adam, de *Adaptive Moment Estimation*, que tal como outros algoritmos, é um algoritmo de aprendizagem adaptativa que usa o calculo de gradientes similares ao *momentum* e da variância desses gradientes.

Se abordarmos as redes neuronais mais detalhadamente, consegue-se perceber a complexidade que estas acarretam. São incluídas análises com a função *sum-of-squares error*, método da entropia cruzada, funções de ativação diferentes, etc. Tudo se resume a um objetivo, obter o melhor regressor/classificador com o menor número de iterações possível e consequentemente no menor tempo de calculo que possível. Existem várias formas de se escolher a função de ativação, o número de camadas escondidas, o número de neurónios de cada camada escondida, quais os indicadores ou atributos corretos, qual o método de determinação de propagação do erro, entre outros parâmetros. Por exemplo, um método heurístico definido por Kirk [14] para a escolha de quantos neurónios se devem escolher segue os seguintes tópicos:

- O número de neurónios nas camadas interiores deve pertencer ao intervalo entre o número de neurónios de entrada e o número de neurónios de saída;
- O número de neurónios nas camadas interiores deve ser igual a 2/3 do número de neurónios de entrada somado do número de neurónios de saída;
- O número de neurónios nas camadas internas deve ser menor que o dobro do número de neurónios de entrada.

A seguinte imagem ilustra a comparação entre o resultado da separação de dois grupos de dados através de redes neuronais artificias com diferentes números de camadas escondidas.

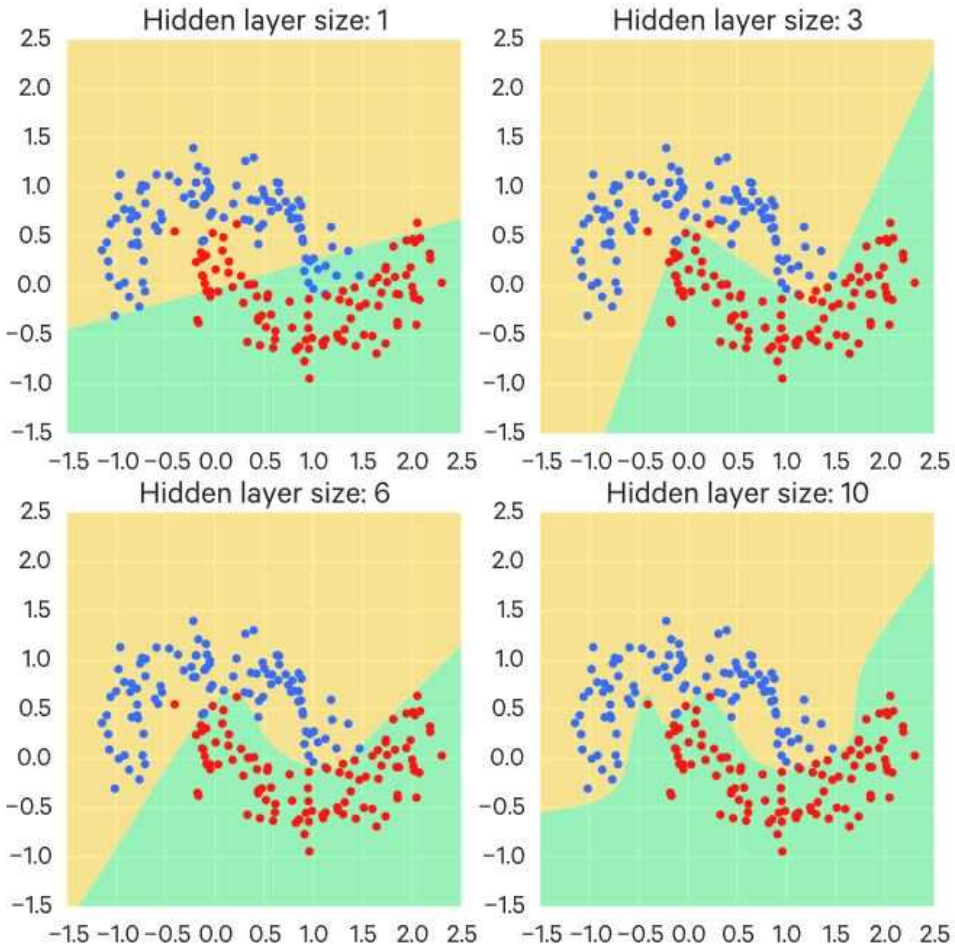


Figura 3-16 - Rede neuronal complexa, função complexa[9]

Na Figura 3-16 pode-se verificar que quanto maior a complexidade da rede neuronal, face apenas ao número de camadas escondidas, maior é a capacidade na distinção de dados de diferentes classes.

Escrever um algoritmo de raiz é uma boa maneira de perceber os princípios fundamentais de como e porque é que um dado algoritmo preditivo funciona, mas por vezes, desta forma, não se obtém a eficiência desejada. Partindo de algoritmos já elaborados, combinando-os ou modificando-os de forma a obter melhores resultados face a um dado problema permite uma otimização do tempo.

Dentro da comunidade de programadores em *Python*, existe uma diversidade de ferramentas/bibliotecas dedicadas à análise de dados, tratamento e previsão. A biblioteca mais frequentemente usada é a Scikit-learn, e é nesta que se baseiam todos os testes de aprendizagem de máquina elaborados neste projeto.

O Scikit-learn[17] é uma biblioteca/módulo em python que fornece vários algoritmos de aprendizagem supervisionada e não supervisionada. Foi construída sobre os módulos bastante familiares na programação python tais como o NumPy, Pandas e o Matplotlib.
Algumas das funcionalidades que se podem encontrar no módulo do Scikit-learn são[18]:

- Regressões, lineares e logísticas
- Classificações, tais como o método K-Nearest Neighbors
- Agrupamentos (Clustering), com o método K-Means e K-Means++
- Seleção do modelo de aprendizagem de máquina
- Pré-processamento de dados, tal como a normalização Min-Max (ou softMax)

Uma dessas funcionalidades é o classificador perceptrão de multicamada (MLPC) que usa como algoritmos: Adam, L-BFGS e a descida de gradiente com aprendizagem estocástica (SGD - *Stochastic Gradient Descent*). A formulação genérica tem a expressão seguinte:

$$w \leftarrow w - \eta \left(\alpha \frac{\partial R(w)}{\partial w} + \frac{\partial Loss}{\partial w} \right)$$

Onde:

- α é um termo que ajuda a evitar classificação grosseira penalizando pesos com grandes magnitudes;
- η é o ritmo de aprendizagem;
- w é o peso;
- Loss é a função do erro ou também chamada de função de custo

O algoritmo de Adam é semelhante ao SGD porque é um optimizador estocástico. Este ajusta automaticamente o quanto necessita de atualizar nos parâmetros, baseando-se em estimativas de momentos de 1ª e 2ª ordens.
L-BFGS é um método que aproxima a um mínimo local recorrendo à matriz de Hessian, que representa as derivadas parciais de segunda ordem da função de custo. Este atualiza os parâmetros recorrendo à matriz de Hassian inversa. Ao contrário do Adam e SGD, este método não permite a "aprendizagem por minigrupos" (*mini-batch learning*). A aprendizagem por minigrupos consiste na separação dos dados de treino em pequenos grupos por forma a acelerar a convergência a um mínimo local na função de custo.

3.4 Análise de vibrações

Os rolamentos são aplicados em quase todo o tipo de máquinas rotativas. Com a evolução/melhoramento do processo de fabrico e materiais utilizados, o tempo de vida face à fadiga do rolamento não é de todo o principal fator de falha em serviço. Em vez disso, várias falhas prematuras nos rolamentos, podem ocorrer devido desequilíbrio, desalinhamento, rugosidade na superfície, imperfeições geométricas, defeitos discretos, contaminação e temperatura extrema.

Quando se começa a desenvolver uma falha nos rolamentos, resultam pulsos de vibração conhecidos por frequências dos rolamentos. Sendo que, um aumento acentuado na banda de alta-frequência dessas vibrações pode aparecer antes deste se queimar.

Segundo Lin, Huang e Su [19], alguns dos métodos utilizados na determinação de falhas prematuramente são baseados em vibração, acústica, temperatura e partículas metálicas na análise ao óleo de lubrificação. Mas, de todos esses métodos, as medidas de vibração com foco na análise espectral é a abordagem mais usual.

Existem diferentes métodos de análise espectral, tais como ***D**iscrete **F**ourier **T**ransform* (DFT), ***F**ast computation algorithm for Discrete **F**ourier **T**ransform (FFT)*, ***s**hort-**t**ime **F**ourier **T**ransform* (STFT), **W**igner – **V**ille ***d**istribution* (WVD), etc. A maioria dos analisadores espectrais de vibração seguem métodos melhorados FFT ou mesmo o algoritmo FFT generalizado, na equação (3.12)[19].

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\left(\frac{2\pi}{N}\right)nk} (k = 0,1,2, \dots, N-1) \tag{3.12}$$

Onde se assume que o sinal segue o teorema de Dirichlet, em que x(n) é a amostragem no domínio do tempo, X(k) é a amostragem no domínio da frequência, N é o tamanho da série, $e^{-j2\pi(k.n/N)}$ é a soma do cosseno e seno, e k são os múltiplos da frequência fundamental.

De certa forma, a transformada de Fourier calcula a correlação entre o sinal e as ondas sinusoidadais/cosinusoidais que o constituem[20].

Com a transformação do sinal para o domínio da frequência ao em vez do tempo como na Figura 3-17, são mais visíveis os efeitos que um rolamento degradado causa na vibração visto que maioritariamente estes fenómenos são cíclicos[21].

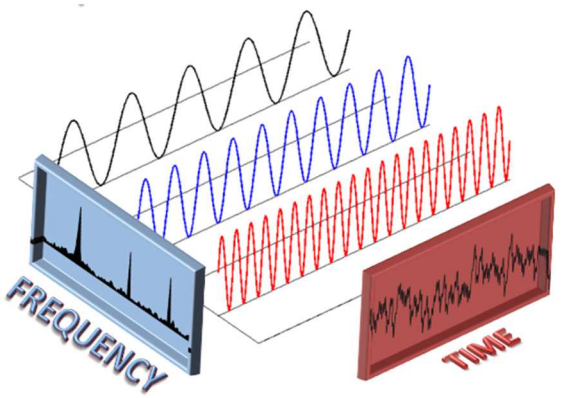


Figura 3-17 - Fast Fourier Transform [20]

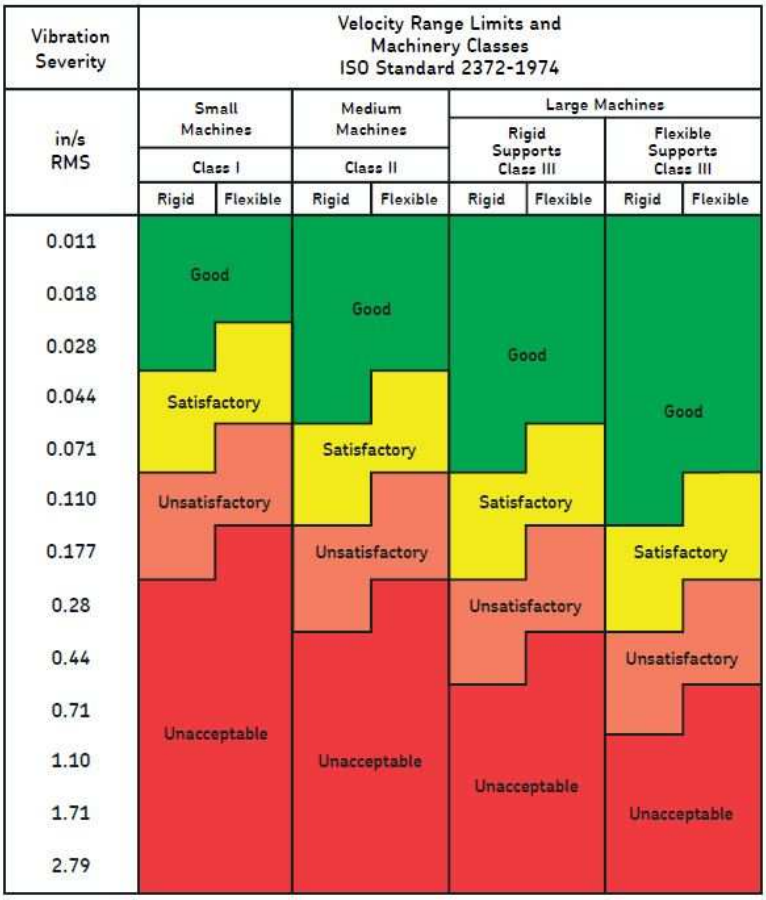
Outra ferramenta que complementa o estudo do espectro de um sinal é a distorção harmónica total, *Total Harmonic Distortion* (THD), que verifica o quão distorcido, face à frequência fundamental, está um sinal, que o impede de ser uma sinusoide perfeita de uma dada frequência [22]. Este cálculo recorre às amplitudes da frequência fundamental e dos respetivos harmónicos na (3.13), tendo em conta que o limite do somatório pode ser qualquer outro acima de 12 desde que o sistema consiga medir esses harmónicos.

$$THD_F = \frac{\sqrt{\sum_{i=2}^{12} A_i^2}}{A_1} \tag{3.13}$$

Em que (A) é a amplitude de cada frequência definida pelo índice (i). A letra (F) em índice de THD é usada para distinguir THD fundamental (F) de THD *root mean square* (R), que tem uma formulação diferente.

De acordo com a SKF, uma das medidas comuns na análise de vibração, é a velocidade. Este tipo de medida, é bastante útil na deteção de problemas rotacionais a baixas frequências, tais como *desequilíbrio*, desalinhamento, veio deformado (numa determinada direção), componentes soltos, etc. A Tabela 3-1 ilustra a classificação ISSO 2372 Standad de acordo com a severidade da vibração.

Tabela 3-1 - ISO 2372 Standard



Nesta tabela podem ser destacados alguns aspetos. É dividida por 4 estados conservação o que se reflete num ponto de partida para classificar um determinado padrão. Embora a dimensão da máquina varie, as classes acompanham de forma linear, querendo isto dizer que uma classificação treinada com dados num motor de pequenas dimensões pode ser transformada para um motor de dimensões maiores através uma extrapolação.

Um exemplo de análise espectral pode ser observado na figura seguinte, se verifica uma elevada amplitude para a frequência de 70 Hz ou 4237.5 RPM e também uma indicação de um desalinhamento no pico de amplitude menor com a frequência de 141 Hz ou 8475 RPM.

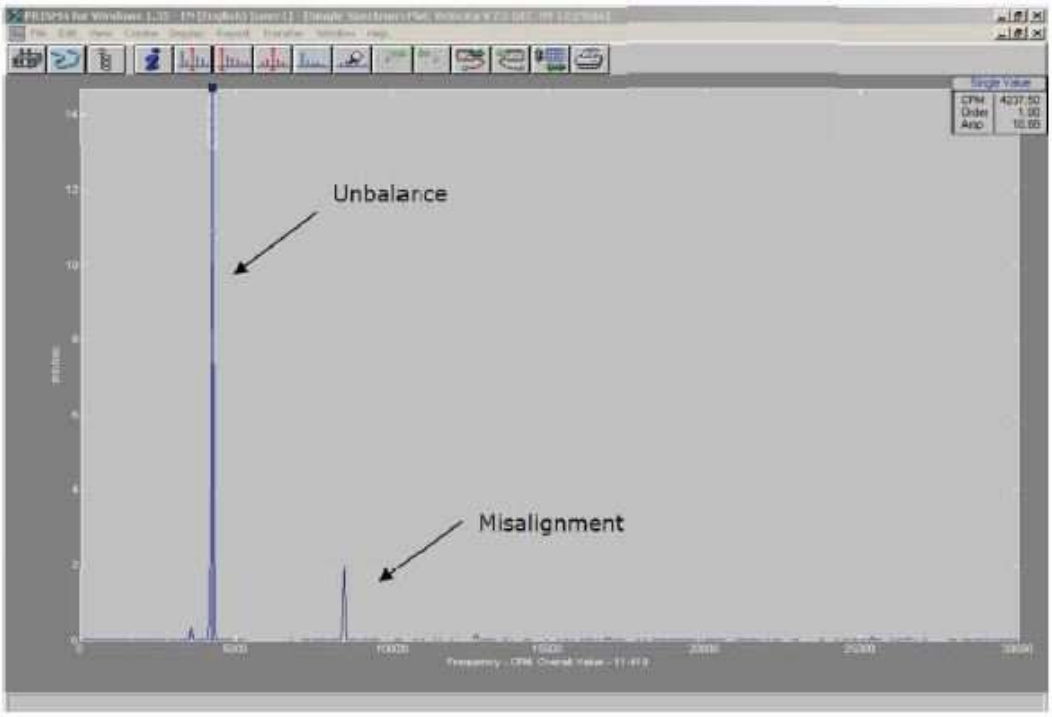


Figura 3-18 - Gráfico FFT de desequilíbrio (*Unbalance*) e desalinhamento (*Misalignment*)

Para além da frequência fundamental e dos respetivos harmónicos devem ser analisadas as várias frequências inerentes dos rolamentos, causadas pelos diversos constituintes assinalados na Figura 3-19.

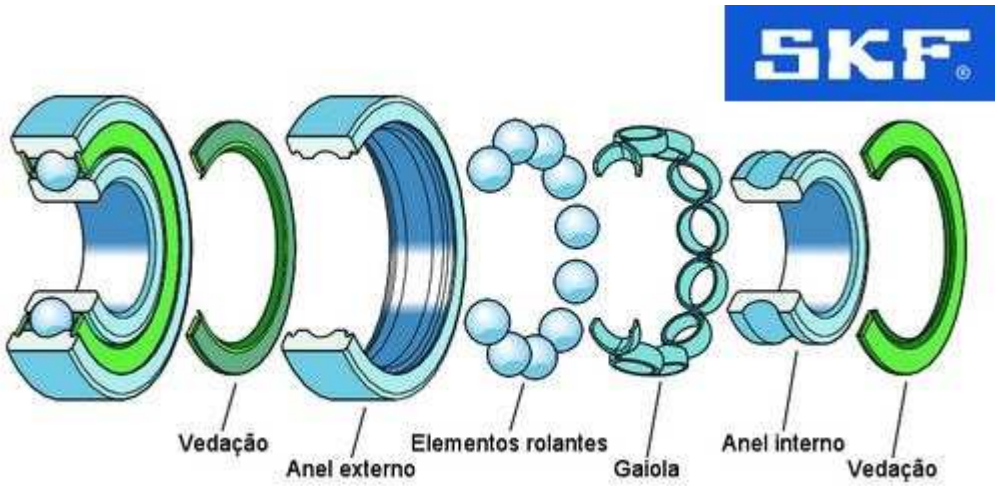


Figura 3-19 - Partes constituintes de um Rolamento [23]

Essas frequências são definidas pelas 4 equações seguintes (Graney,K Starry) [24]:

$$BPFI = \frac{N}{2} \times F \times \left(1 + \frac{B}{P} \times \cos(\theta)\right) \tag{3.14}$$

$$BPFO = \frac{N}{2} \times F \times \left(1 - \frac{B}{P} \times \cos(\theta)\right) \tag{3.15}$$

$$FTF = \frac{N}{2} \times \left(1 - \frac{B}{P} \times \cos(\theta)\right) \tag{3.16}$$

$$BSF = \frac{P}{2B} \times F \times \left[1 - \left(\frac{B}{P} \times cos(\theta)\right)^2\right] \tag{3.17}$$

Onde,

BPFI = frequência de passo dos elementos rolantes na pista interior em Hz

BPFO = frequência de passo dos elementos rolantes na pista exterior em Hz

FTF = frequência fundamental em Hz ou também chamada de frequência de gaiola

BSF ou BPF = frequência de passo dos elementos rolantes

N = número de elementos rolantes

F = velocidade de rotação do veio em rotações por segundo ou o mesmo que dizer em Hz

B = diâmetro dos elementos rolantes

P = diâmetro do rolamento, medido entre centros de elementos rolantes

Θ = ângulo de contacto dos elementos rolantes

Tendo em conta as equações(3.14), (3.15), (3.16) e (3.17), de um modo generalizado, a degradação dos rolamentos pode ser classificada em quatro estados:

1. É recomendada a monitorização no intervalo normal estabelecido pelo fornecedor.

   É visível um aumento significativo particularmente na BPFO. Isto deve-se ao facto de na maioria das montagens as máquinas estarem montadas na horizontal ficando rolamento na perpendicular com a força gravítica e por esse motivo criar um maior desgaste no anel exterior.

2. Recomenda-se a troca dentro do intervalo normal de manutenção preventiva.
   São visíveis aumentos nas amplitudes dos harmónicos das frequências de falha de um modo geral, embora que este efeito se reflita para a BPFO no primeiro harmónico e para BPFI no segundo.
3. Recomenda-se a troca dentro dos próximos 30 a 45 dias.
   Com amplitudes ainda maiores, as BPFO e BPFI são bastante distintas incluindo os respetivos harmónicos. Pode ainda ser encontrada a BSF com aumento significativo particularmente no primeiro harmónico.
4. É recomendada a troca imediata.
   Neste estado de degradação, começam a aparecer frequências de vibração aleatórias, aumentos no segundo e terceiro harmónico da frequência da velocidade de rotação do veio que se podem traduzir em componentes soltos, como é o caso da *Figura 3-18*. Caso apareça saliente a FTF, significa que o rolamento está na iminência do fim de vida.

Claro que para obter medidas rigorosas é necessário ter um bom suporte físico, e no que toca a esse respeito, existem inúmeras soluções no mercado para sensores de vibração. Estas soluções podem variar entre os 25 € e os 1700 € [25]. Embora o equipamento mais barato, SensorTag [26], possa ser limitado face aos restantes equipamentos, foi concebido com enfoque no conceito de IIoT (Secção 3.1.1) o que permite a entrada de pequenas e médias empresas na Industria 4.0, mais propriamente na manutenção preditiva e preventiva.

3.5 Influência da Temperatura no tempo de vida de rolamentos

Apenas como referência, e também porque o tipo de rolamentos a analisar será de esferas, esta secção exclui os rolamentos cilíndricos, cónicos e de contacto.
Com base na *Aplication Note* da JIB [27], a gama típica de operação compreende os -20ºC a 100ºC, sendo que se pode trabalhar com temperaturas superiores mediante a massa lubrificante aplicada. O aumento de temperatura é causado pela transferência de calor durante a rotação dos rolamentos e pela fricção e resistência que a massa lubrificante confere. Como a própria rotação do rolamento dissipa calor, normalmente, atinge o máximo de temperatura ao fim de 30 min a 2 horas de trabalho, chegando a um equilíbrio quando se mantêm numa temperatura 3 a 5ºC abaixo da máxima. Esta estabilização de temperatura ao fim de um certo tempo deve-se à quantidade de massa lubrificante chegar a um ponto ótimo, isto é a quantidade suficiente que confere maior deslizamento aos elementos rolantes, mas não em demasia que comece a contribuir para o atrito dos mesmos.

Com estes tópicos pode ser definida uma estimativa de temperatura no arranque de uma máquina e com base na medição contínua verificar se a tendência se vai afastando dessa estimativa e com que ritmo e quanto é que esse aumento contribui para a determinação do estado dos rolamentos.

3.6 Soluções comerciais atuais

Atualmente várias empresas estão a aderir à Indústria 4.0 quase como um novo standard exigido pelas empresas consumidoras. Algumas das soluções existentes são, a COMOS da Siemens, FIELD da FANUC e o WATSON da IBM

A aplicação ZDT (*Zero Down Time* – tempo de paragem nulo), que corre no sistema da FANUC, FIELD, adquire dados de mais de 6000 robôs em 26 fábricas salientando a importância do conceito de Big Data. A ZDT monitoriza esses robôs para encontrar desgastes anormais que possam provocar avaria do equipamento. Caso esteja nessas circunstâncias, é emitido um aviso, sob forma de email ou outros meios, antes que a paragem de uma linha de produção ocorra. Com a ZDT, o FANUC rastreia o uso do robô e envia lembretes com intervalos apropriados por forma a garantir que a manutenção é feita a tempo [2].

A Siemens tal como a FANUC, tem uma abordagem por aplicações. O COMOS, a solução apresentada por esta empresa, divide-se em:

- COMOS Automation
- COMOS Lifecycle
- COMOS MRO (Maintenance, Repair & Overhaul)
- COMOS Platform

O COMOS Lifecycle, que é o tópico mais importante de se mencionar, embora baseado no conceito de manutenção preditiva com a análise e diagnóstico de uma base de dados em tempo real, leva-o um passo à frente, comprometendo-se a ser extensível a uma fábrica inteira e não apenas a uma secção [3]. Na expanção da The Navigator Company em Cacia, com a construção de uma nova fábrica (Navigator Tissue Cacia), foram já adotados alguns sistemas da Siemens com tecnologia COMOS Lifecycle, no que toca a monitorização de vibrações e temperatura de um equipamento vital para o processo de formação de papel Tissue. Este equipamento consiste num rolo com 122 toneladas, 5,5m de diâmetro e 6,3m de comprimento que pode rodar até 2 rotações por segundo, o que implica uma monitorização bastante rigorosa que inclui trasnformadas de Fourier nos gráficos de vibração.

O Watson da IBM, na sua vertente de IoT, é o software com que o tema desta dissertação se identifica. Defendem que numa fábrica a média de softwares diferentes devido às marcas diversificadas chega

a ser 150 e que muitas vezes impede a sinergia harmoniosa no sentido da IIoT. Além de permitir a construção e novos modelos preditivos usando a plataforma Watson, permite também usar os modelos desenvolvidos pela IBM e melhorá-los adaptando-os aos dados recolhidos, com recurso à ferramenta “IBM Maximo APM - Predictive Maintenance Insights”[28].

4 Implementação e análise de desempenho

O diagrama presente na Figura 4-1 ilustra o sistema montado e a direção do fluxo de informação desde a aquisição de dados à previsão do tempo de vida de um equipamento, nomeadamente um motor. Nesta figura podemos ver várias etapas numeradas de 0 a 5, que são abordadas com detalhe nas secções seguintes.

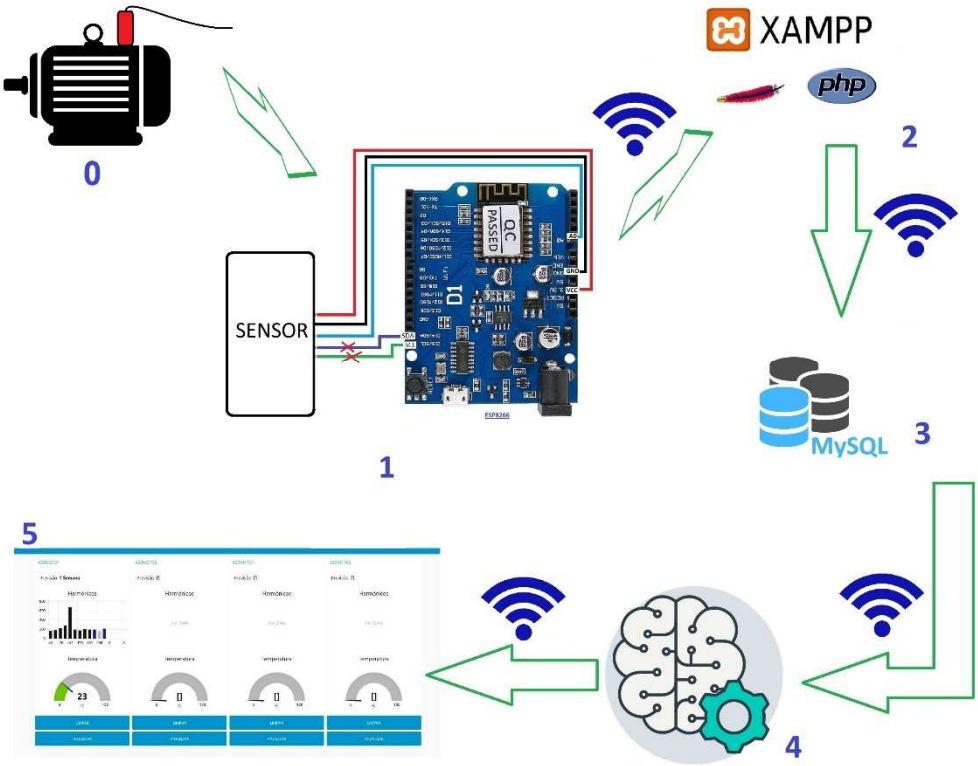


Figura 4-1 – Diagrama do Projeto

A etapa 0 ilustra a aplicação do sensor num motor como se poderá constatar posteriormente na secção 4.1. Esta apresenta detalhes sobre a obtenção de valores de aceleração e respetivas frequências obtidas em tempo real, em ambiente fabril, bem como as principais características dos objetos de teste. A etapa 1 é composta por um microcontrolador (ESP8266) e pelo sensor ADXL335 tendo ainda sido desenvolvido algum trabalho que consistia na elaboração de um sensor de baixo custo presente nos Anexo I e II. Nesta etapa, além da aquisição de dados, são também extraídas as frequências com

recurso ao algoritmo FFT. A programação do microcontrolador ainda da etapa 1, a configuração e programação do interface com a base de dados (etapa 2) e configuração/organização da base de dados (etapa 3), são abordadas na secção 4.2. A etapa 4 é detalhada na secção 4.4, onde é desenvolvido e implementado o modelo de *machine learning*. A quinta e última etapa diz respeito às 2 interfaces gráficas desenvolvidas, uma de visualização dos resultados e outra para parametrização do sistema, abordadas na secção 4.3.

4.1 Aquisição de dados reais

Atendendo à disponibilidade da secção de produção de pasta para realizar testes sem de alguma forma impedir o funcionamento normal, ao invés dos locais mencionados na secção 2.2, foram medidas as frequências de vibração em motores na área de recuperação e energia, mais concretamente na Evaporação. A secção de produção de pasta durante o período de recolha de dados teve alguns problemas e paragens de produção planeadas que implicam a movimentação de maquinaria, cargas pesadas e sobre-te pessoas Um dos exemplos de aplicação é visível na Figura 4-2 onde estão assinalados por zonas a vermelho, a caixa com IP65 onde se encontra o microcontrolador (circulo da esquerda) e também o local de aplicação acelerómetro no motor (circulo da direita).

Figura 4-2 – Medição de frequências de vibração de um motor em ambiente industrial

Foram obtidas frequências de vibração de três motores cujas características relevantes são apresentadas na tabela seguinte.

Tabela 4-1 - Características dos motores analisados

Nome	**Velocidade [rpm]**	**Potência [KW]**	**Nº linhas de dados obtidos**	**Diagnóstico**
Diag1	1468	15	6610	Ruído anormal devido a má montagem do acoplamento no veio
Diag2	1468	15	2629	Sem avaria
Diag3	1485	55	1059	Rolamentos com degradação avançada

A coluna de diagnóstico, na Tabela 4-1, foi fornecida por técnicos da Empresa SKF, entidade responsável pela monitorização da degradação dos rolamentos em equipamentos de toda a fábrica. Esta monitorização é contínua, mas não é em tempo real visto que se baseia em rotas com diferentes períodos de medição. Isto é, define-se um grau de vigilância numa escala, de por exemplo, 1 a 4, sendo que esse grau de vigilância tem a ver com o estado de degradação dos rolamentos e reflete-se no número de vezes que estes são monitorados numa semana. Desta forma, atribuindo um grau de vigilância ao motor, é inclui-lo numa "rota" que tem um número associado de medições por semana. A cada motor foi atribuído um nome segundo o diagnóstico atribuído, Diag1, Diag2 e Diag3, tendo em conta que o Diag1 e Diag2 são motores iguais e com cargas semelhantes. Não seria intuitivo nem traria qualquer informação relevante, usar o verdadeiro *tag-name* dos Motores. O número de linhas de dados é um conjunto de várias medições em dias diferentes e com variações de carga, enviadas pelo microcontrolador. Existiu esta preocupação para que a variação de carga não se refletisse no espectro de medições e contribuísse erradamente para o padrão identificativo da avaria. Estas linhas têm a estruturação apresentada na tabela seguinte. Sendo esta um pequeno excerto dos dados recebidos no dia 12/10/2018 na medição no motor Diag1.

Tabela 4-2 – Excerto da tabela de dados reais medidos em ambiente industrial do motor Diag1

idrawdata	hora	temperatura_1	f0	m0	f1	m1	f2	m2	f3	m3	f4	m4
874	2018-10-12 10:21:46.000	23.4	73.00	117.19	100.00	415.19	275.00	134.41	300.00	195.79	325.00	225.76
875	2018-10-12 10:22:01.000	23.4	100.00	386.79	125.00	114.71	300.00	156.68	324.00	182.58	0.00	0.00
876	2018-10-12 10:22:17.000	23.4	101.00	265.03	125.00	140.78	145.00	134.31	275.00	146.08	301.00	173.32
877	2018-10-12 10:22:33.000	23.4	100.00	351.38	125.00	194.35	145.00	130.41	276.00	130.52	300.00	207.51
878	2018-10-12 10:22:49.000	23.4	100.00	542.26	124.00	191.06	276.00	149.73	300.00	215.14	325.00	250.30
879	2018-10-12 10:23:05.000	23.4	101.00	253.97	275.00	148.22	300.00	211.15	325.00	152.06	0.00	0.00
880	2018-10-12 10:23:21.000	23.4	100.00	523.48	300.00	218.14	325.00	140.52	0.00	0.00	0.00	0.00
881	2018-10-12 10:23:37.000	23.4	101.00	265.86	276.00	155.96	300.00	195.48	325.00	251.65	0.00	0.00
882	2018-10-12 10:23:53.000	23.4	100.00	377.51	121.00	103.60	145.00	101.01	276.00	143.34	300.00	226.10
883	2018-10-12 10:24:09.000	23.4	100.00	425.99	125.00	120.56	275.00	103.75	300.00	194.58	324.00	149.88
884	2018-10-12 10:24:24.000	23.4	101.00	206.66	276.00	103.96	300.00	163.23	324.00	186.22	0.00	0.00

Na Tabela 4-2 são visíveis algumas das variáveis enviadas pelo microcontrolador, em que cada uma é uma coluna. A primeira coluna (à esquerda) diz respeito ao ID da linha, isto é, ao número que a identifica. A coluna “hora” é um *label* do tipo “TIMESTAMP” que regista a hora a que foram escritos os dados na tabela. A coluna “temperatura_1” é uma das variáveis enviadas pelo microcontrolador e diz respeito à temperatura do sensor. As restantes colunas são as várias frequências, numeradas de 0 a 17, determinadas pelo algoritmo de FFT no microcontrolador, onde, por exemplo, f0 é valor de frequência e m0 é a respetiva amplitude. No caso de num determinado momento, o microcontrolador determinar apenas quatro frequências nas medições de aceleração, são enviadas as quatro frequências, numeradas de 0 a 3 incluindo as respetivas amplitudes. Todos os outros valores, nesse dado intervalo de tempo, de frequências e amplitudes são preenchidos por “0”.
Ainda na Tabela 4-2, verificam-se valores pontuais de amplitudes elevadas, isto deve-se a vários motivos, entre estes, a variação de carga na bomba acoplada ao motor, problemas de processo, fecho de repentino de válvulas que criam golpes de Ariete, etc.

Foram adquiridos 3 acelerómetros ADXL335 com placa de acondicionamento de sinal incluída (vulgarmente usados em projetos didáticos com Arduinos, canto inferior direito Figura 4-3), que foram colados em motores sendo apenas lido o sinal analógico do eixo Z, isto é, perpendicular ao eixo de rotação do rolamento. O acelerómetro foi colado a uma placa de baquelite para evitar curto-circuitos e por sua vez esta foi colada diretamente ao equipamento a medir e por este motivo não pôde ser recuperado. Todas as ligações expostas ao ambiente, foram protegidas com tubo termo retrátil. O facto de não se usar o módulo expansor de entradas analógicas impossibilitou a medida de temperatura (e dos outros eixos de medida da aceleração) porque o microcontrolador usado apenas tem uma entrada analógica, tendo esta ficado forçada no microcontrolador. Todo o software assume

que ela existe e é incluída como um indicador do estado do rolamento para posteriores cálculos relacionados com o *machine learning*. O equipamento usado em ambiente fabril resume-se ao acelerómetro e ao microcontrolador visíveis na Figura 4-3, claro juntamente com um router e um computador (servidor).

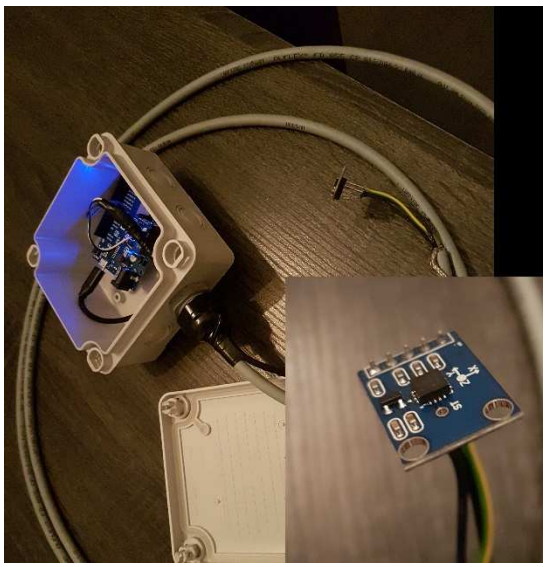


Figura 4-3 - Equipamento montado para medição de vibrações

O acelerómetro usado (ADXL335) possui uma taxa de amostragem de 1600Hz, uma gama de medida entre os -3 e 3g e uma resolução de 12bit e pode ser adquirido por pouco menos de 3 euros. Comparativamente aos outros sensores do mercado, na Tabela 4-3, significa uma oportunidade de desenvolvimento a baixo custo. Foi efetuado um estudo a nível de circuito elétrico e involucro com o intuito de desenvolver um dispositivo que medisse aceleração e temperatura a baixo custo, contudo não foi possível continuar esse desenvolvimento ficando disponível para consulta nos Anexos I e II.

Tabela 4-3 - Comparação entre sensores de vibração

Nome do Produto	**SensorTAG**	**VB300**	**Vibration Meter**	**MSR165**	**SlamStick**	**LabVIEW System**
Custo	$29	$210	$1935	$1271 a $2956	$1000 a $2750	$6250

Taxa de amostragem (Hz)	10	200	20000	1600	20000	51200
Gama de Medida (g)	16	18	50	15 ou 200	16, 25, 100, 500 ou 2000	Várias Opções
Armazenamento (Milhões de amostras)	0	0,50	0,004	2	1000	Ilimitado
Resolução	16 bit	12 bit	16 bit	13 bit	16 bit	24 bit

4.2 Programação do microcontrolador e interligação com a base de dados

O microcontrolador foi programado em linguagem C com o software Arduino IDE. Neste a são feitas a aquisição da aceleração, da temperatura por meio de uma leitura de uma entrada analógica (ADC - *analog to digital converter*) e determinadas as frequências de vibração com recurso a uma biblioteca do algoritmo da transformada de Fourier descrito na secção 3.4. Este algoritmo pressupõe a configuração de vários parâmetros, desde a frequência de amostragem ao número de amostras. A frequência de amostragem é de 1 KHz, frequência esta que se fosse menor, possibilitava a medição de frequências mais baixas com melhor precisão e se fosse demasiado alta, não funcionava devido ao à aproximação à frequência máxima do ADC (10 KHz). Não se justifica usar um valor superior a 1 KHz porque as frequências são maioritariamente ruido. A resolução da ADC do microcontrolador é de 10 bits, com uma tensão de referencia de 3,3 V o que corresponde a aproximadamente 3,22 mV/bit. A saída do sensor tem a mesma excursão de sinal que a ADC do microcontrolador, não sendo necessário fazer o seu acondicionamento, para alem daquele que as placas já possuem. São usadas 512 amostras para o cálculo do FFT, e o tipo de janela aplicada é a de Hamming[35]. Foram definidos dois atrasos entre cada ciclo do programa, um de cinco segundos para o sistema fazer uma média dos valores medidos durante o arranque do sistema e registá-los, para subsequentemente, usa-los como offset calibrando o “zero” do acelerómetro. O “zero” do acelerómetro é necessário devido às vibrações de fundo existentes na estrutura, mesmo quando a máquina está parada. O segundo é o

intervalo de tempo entre cada ciclo (10 s). Podendo estes tempos serem ajustados conforme a importância da equipamento/máquina a medir (criticidade).

Foram feitos alguns ajustes na medição da frequência com recurso a uma mesa vibratória de velocidade fixa nos 40 Hz e um osciloscópio PicoScope 2000 [36]. Esses ajustes consistem em duas operações. Uma foi uma média ponderada para atenuar um pouco a imprecisão no calculo das frequências, e a outra operação, resume-se a um filtro por software que exclui todas os valores de frequências cujas amplitudes sejam menores que um determinado valor (neste caso 100). O resultado destas operações pode ser constatado na Figura 4-4, onde o gráfico à direita corresponde ao antes e o da direita corresponde ao depois. Obtendo o resultado na Figura 4-4, do lado esquerdo.

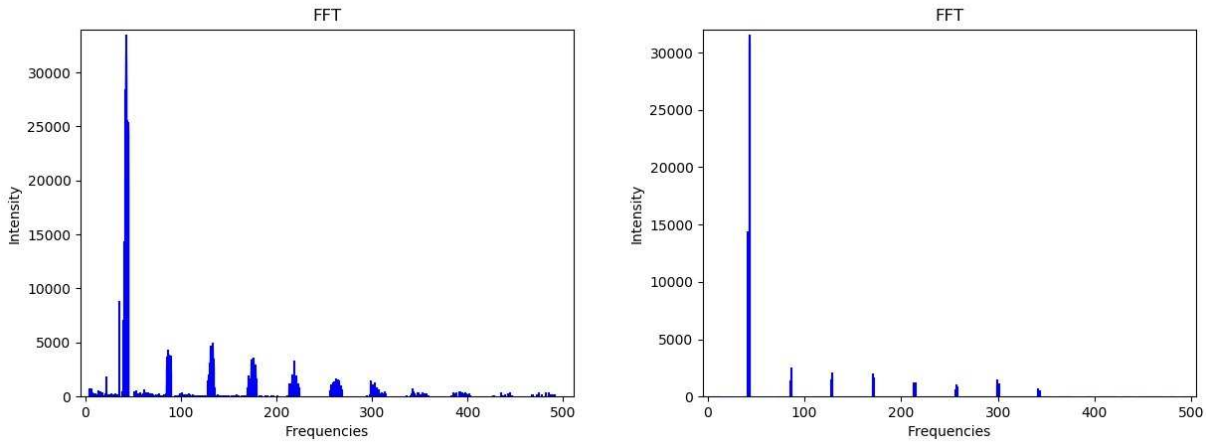


Figura 4-4 - Comparação da medição FFT

Segundo a solução proposta apresentada

+ no início da secção 4, a 2ª etapa consiste num pacote de serviços denominado de XAMPP[37], Deste foram usados um servidor Apache e um servidor MySQL. O Apache é um servidor do tipo HTTPD compatível com protocolo HTTP 1.1 que permite o acesso a ficheiros (HTML, PHP, SQL, entre outros), alocados na mesma máquina em que corre o servidor, por ligação TCP/IP. O ficheiro acedido em questão é um código PHP desenvolvido para servir de interface entre o microcontrolador e a base de dados. As variáveis, 17 valores de frequência, as respetivas amplitudes e a temperatura, são preenchidas pelo método GET e concatenadas numa *string* que é incluída posteriormente numa *query* de INSERT na base de dados elaborada.

A base de dados (3ª etapa) é composta por sete tabelas como se pode ver na Figura 4-5. As ligações entre Python (software onde correm os algoritmos de *machine learning*) e o NodeRED mostram-se

praticamente impossíveis porque não existe documentação viável que fundamente se é até possível realizar tal ligação. Mas como as ligações entre o script python (Secção 4.4) e MySQL, e NodeRED (Secção 4.3) e MySQL foram conseguidas, a forma de ultrapassar este problema foi criar duas tabelas, denominadas de "display" e "node_red_prev", assinaladas pelo grupo azul na Figura 4-5.

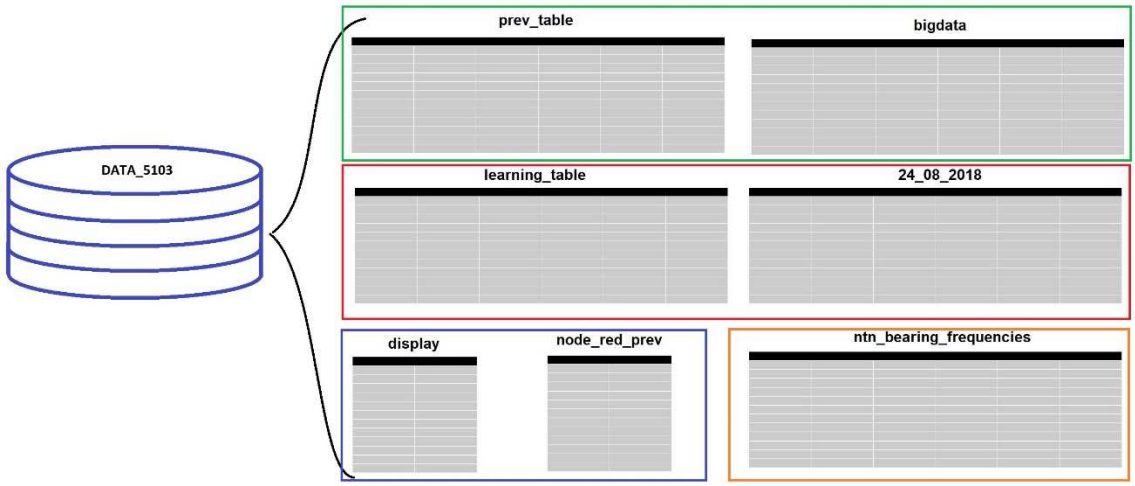


Figura 4-5 - Representação da base de dados

Na Figura 4-5 verificam-se 7 tabelas estando estas divididas por zonas num código de cores tendo sido já apresentado um dos grupos/zona vermelha na Tabela 4-2. Esta classificação por cores é meramente ilustrativa e pretende apresentar as tabelas, isto é, tabelas do mesmo grupo não implica que estas sejam estruturalmente iguais ou que se complementem, mas sim porque têm algo em comum. O grupo verde representa as tabelas de resultados da previsão. No grupo vermelho, estão as tabelas usadas na aprendizagem do modelo de *machine learning*, onde a tabela nomeada com uma data, neste caso, "24/08/2018", é o armazenamento dos dados do microcontrolador. Os dados recebidos do microcontrolador são guardados numa tabela cujo nome é definido pelo programa de parametrização inicial (secção 4.3), que por defeito atribui-lhe como nome o dia em que o programa começa a fazer monitorização contínua. Por fim tem-se a tabela a laranja que foi preenchida com dados, de cerca de 2000 rolamentos diferentes, da Empresa NTN [38] tal como por exemplo a frequência de passo dos elementos rolantes na pista interna de um dado rolamento (BPFI).

Como forma de testar o programa do microcontrolador e o envio das frequências e respetivas amplitudes para a base de dados, foi elaborada uma pequena experiencia com o PicoScope 2000 em modo de gerador de sinais. É importante referir que esta medição foi feita antes da implementação

da média ponderada e filtro na programação do microcontrolador, verificando-se por isso frequências muito próximas na Figura 4-7.

A Figura 4-6 demonstra o sinal gerado pelo PicoScope 2000 em modo de gerador de sinais, com um sinal sinusoidal com a frequência de 60 Hz e 1 V de amplitude. As pontas de prova estavam ligadas positivo com a entrada analógica do microcontrolador e a referência da ponta com a referência do microcontrolador.

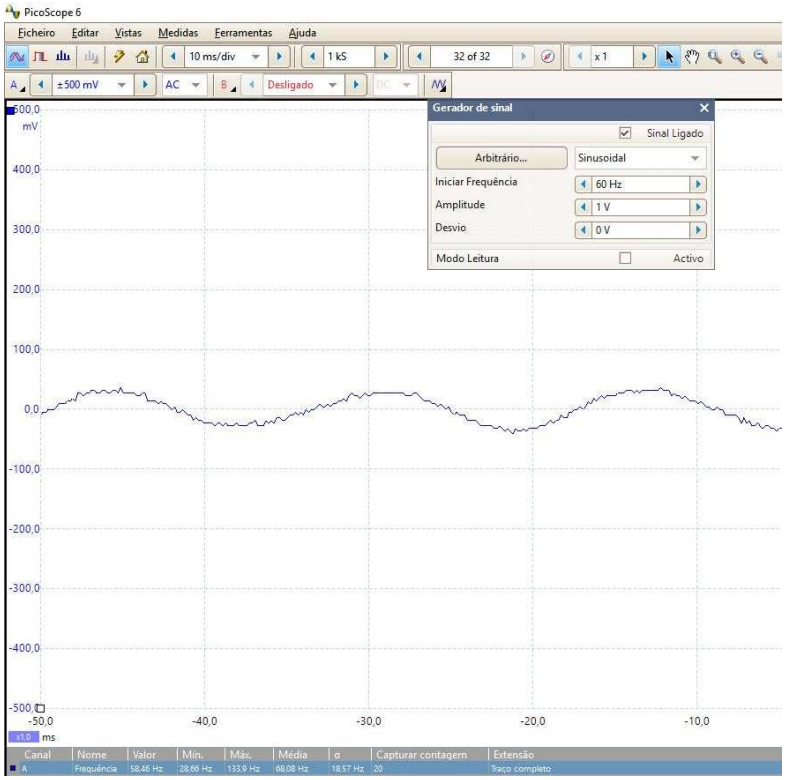


Figura 4-6 – Teste para ajuste na determinação de frequências medidas na ADC do ESP8266

A Figura 4-7 é o resultado da medição do sinal acima descrito, numa tabela, preenchida por valores de frequência e respetiva amplitude. Na tabela além de se constatarem valores de frequência muito próximos como é o caso dos 59, 61 e 63 Hz que necessitaram claramente de um ajuste para que correspondam a 60 Hz, verificam-se medidas do 2º harmónico (120 Hz), o que significa que nãos e trata de um sinal sinusoidal de 60Hz.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	hora	temperatura_1	f0	m0	f1	m1	f2	m2	f3	m3	f4	m4	f5	m5
559	2018-04-19 01:29:58.000	23.4	2.00	263.39	59.00	491.03	61.00	734.38	63.00	182.37	119.00	226.41	121.00	219.63
560	2018-04-19 01:30:14.000	23.4	2.00	261.74	59.00	488.39	61.00	734.72	63.00	184.89	119.00	229.24	121.00	222.45
561	2018-04-19 01:30:30.000	23.4	2.00	265.24	59.00	495.35	61.00	742.00	63.00	188.09	119.00	232.41	121.00	224.01
562	2018-04-19 01:30:46.000	23.4	2.00	271.63	59.00	493.83	61.00	741.61	63.00	191.24	119.00	226.74	121.00	224.24
563	2018-04-19 01:31:02.000	23.4	2.00	265.61	59.00	490.41	61.00	736.48	63.00	186.18	119.00	225.30	121.00	221.03
564	2018-04-19 01:31:18.000	23.4	2.00	270.49	59.00	491.65	61.00	741.72	63.00	189.99	119.00	226.37	121.00	225.81
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 4-7 - Medições da frequência do sinal gerado pelo PicoScope 2000 (60 Hz)

4.3 Interfaces Gráficas

A interface gráfica do sistema SCADA (Figura 4-10) foi desenvolvida com um editor gráfico *web-browser* conhecido por NodeRED. Esta ferramenta começou como um *side-project* de dois colaboradores da IBM, e foi desenvolvido como uma ferramenta de programação para conectar dispositivos de hardware, APIs e serviços online [39].

O NodeRED é um servidor, que no caso deste projeto, corre localmente na máquina onde reside o servidor MySQL e Apache, e permite a ligação por TCP/IP para configuração de uma página web que pode ser acedida também por TCP/IP.

Como se pode ver na Figura 4-8, a programação principal é feita por um fluxograma suportado por *nodes* ou módulos. Da esquerda para a direita, começando no módulo roxo, este foi configurado para mandar um impulso a cada segundo, sendo responsável pela atualização da página web. De seguida tem-se um módulo laranja claro, que é um módulo de função. Este tipo de módulos pode desempenhar inúmeras operações porque são configuráveis por linhas de código em linguagem C ou JSON (*Java Script Object Notification*) como se pode ver na Figura 4-9, à direita. Neste caso este consiste numa *query* SELECT para ler uma determinada tabela da base de dados. A ligação propriamente dita à base de dados é feita pelo módulo laranja escuro e a representação para o *User Interface*, de mensagens, gráficos, entre outros, é feita pelos módulos azulados à direita.

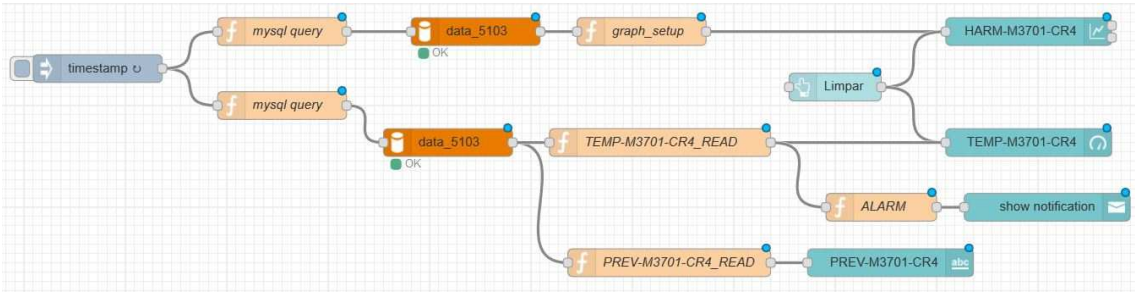


Figura 4-8 - Programação por flow/nodes

Um exemplo de configuração de um dos módulos de representação é visível também na Figura 4-9, mas do lado esquerdo, onde se pretende configurar o aspeto de um manómetro para indicar a temperatura. Estes módulos são providenciados gratuitamente pela comunidade do NodeRED.

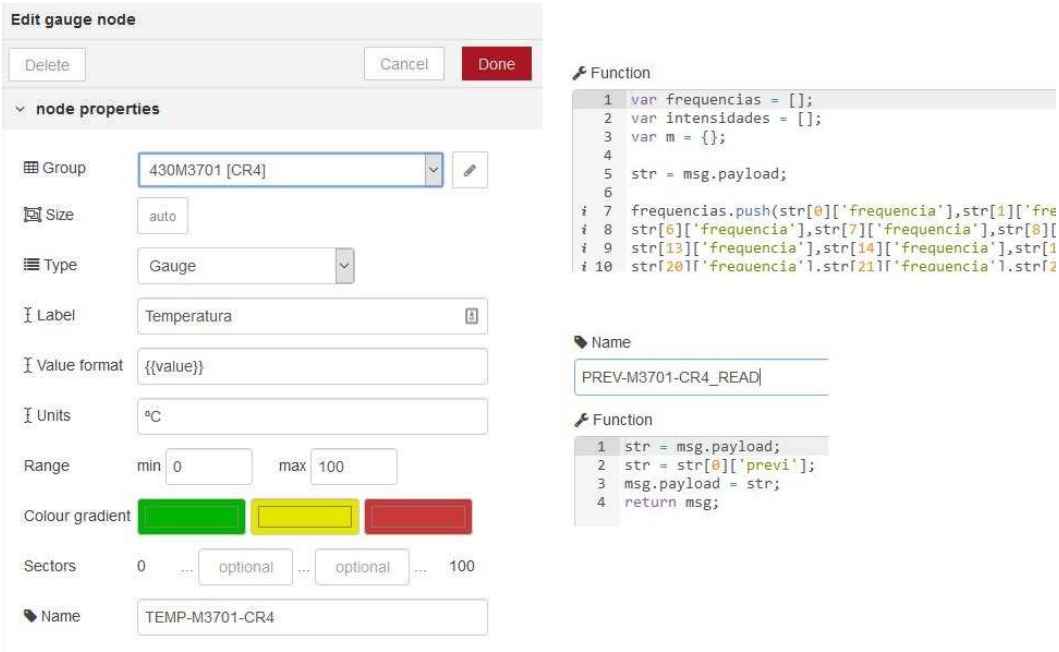


Figura 4-9 - Configuração de módulos, à direita por interface gráfica e à esquerda por programação

A Figura 4-10 ilustra o resultado final de uma página elaborada em NodeRED, onde podemos ver os elementos de forma sucinta. Cada coluna corresponde a um equipamento, identificado com um TAG, sendo que cada coluna é composta pelo último gráfico de frequências medidas nesse equipamento, a temperatura lida pelo sensor e a previsão do tempo de vida. Na programação NodeRED os gráficos necessitam de estar formatados em apenas duas colunas verticais sendo estas portanto lidas da tabela "display" assinalada a azul na Figura 4-5.

Figura 4-10 – SCADA de monitorização

Outra interface gráfica desenvolvida diz respeito à pré-configuração para que o algoritmo de aprendizagem funcione de acordo com as características da máquina. Esta, consiste num programa elaborado em ambiente vb.net com o software Visual Studio 2017 [40] onde se pode verificar o seu aspeto na Figura 4-11.

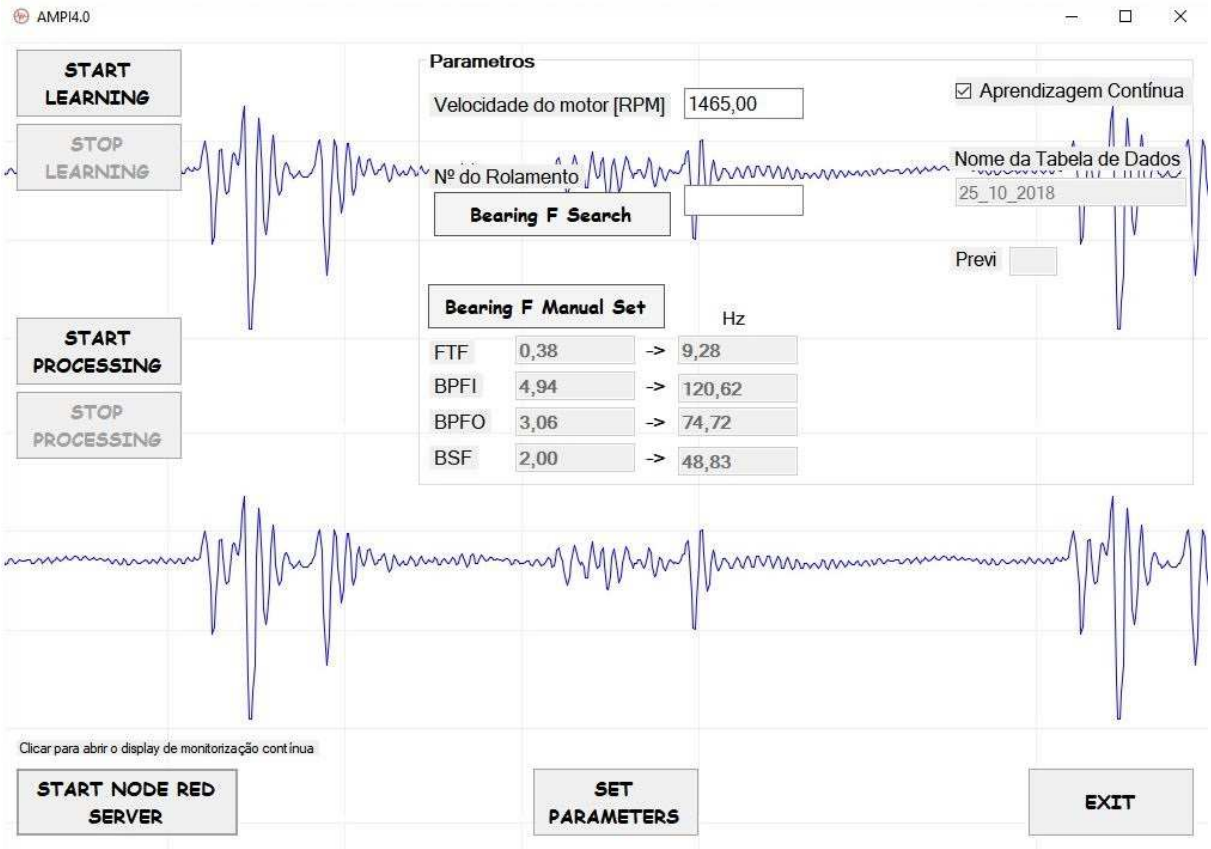


Figura 4-11 - Interface gráfica do programa de configuração do equipamento a medir

Os gráficos visíveis na Figura 4-11 são estáticos e são meramente elucidativos à ideia da monitorização. Uma particularidade destes gráficos é que são medições reais da aceleração durante a fase experimental apenas com o microcontrolador e sensor.

Este programa tem 2 formas de funcionamento, uma é a aprendizagem forçada/manual em que o utilizador dá o nome da tabela de dados de treino juntamente com a previsão do tempo de vida correspondente aos dados de entrada. Para que este modo esteja ativo, deve ser retirado o visto que vem por defeito na *checkbox* "Aprendizagem Contínua". O outro modo de funcionamento é o que vem por defeito selecionado, a aprendizagem automática, e pressupõe a realização de uma sequencia de operações para que o sistema esteja corretamente configurado.

1. Verificar se a *checkbox* "Aprendizagem automática" está selecionada;
2. Verificar se o nome da tabela de dados é o dia atual;
3. Colocar a velocidade de rotação do veio do equipamento a medir (em rotações por minuto);
4. Colocar o número de referência do rolamento mais próximo da medida do sensor;
5. Clicar em **Bearing F Search** para calcular as frequências do rolamento escolhido;
6. Clicar em **SET PARAMETERS** para guardar todas as definições num ficheiro *.cfg;

7. Clicar em **START PROCESSING**;
8. Clicar em **START NODE RED SERVER** para inicializar o servidor, NodeRED e abrir o SCADA no browser.

Caso a tabela acedida para procurar as frequências de um dado rolamento não encontre o número do equipamento a medir, é possível colocar os parâmetros do rolamento manualmente, como se pode ver na Figura 4-12. Sendo posteriormente calculadas as frequências segundo as equações de (3.14) a (3.17) na secção 3.4.

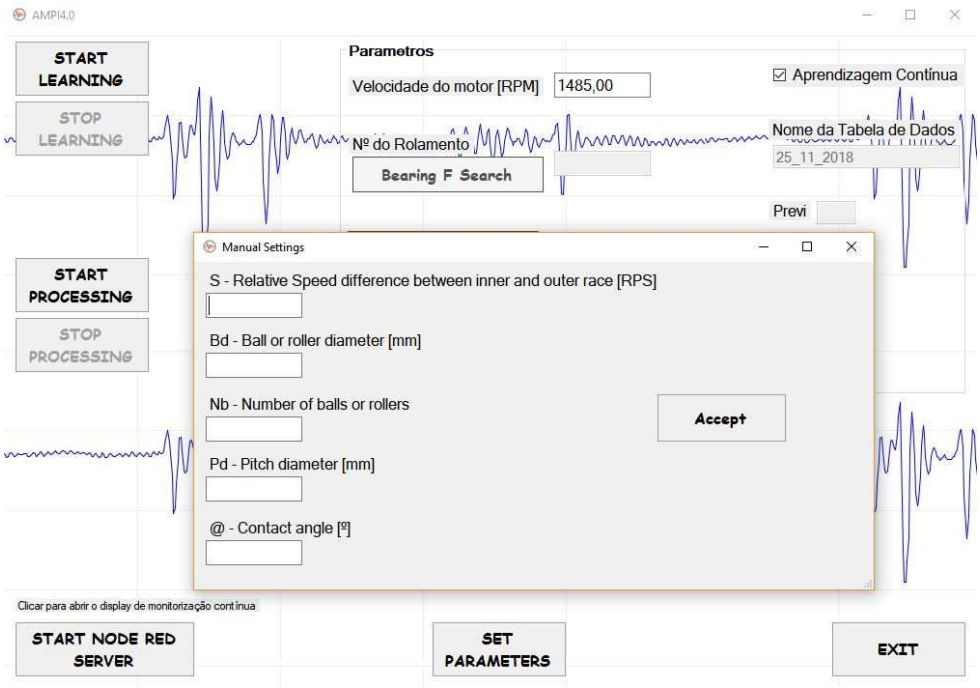


Figura 4-12 - Configuração manual do rolamento

O botão **START LEARNING**, usado no modo manual, é uma forma de melhorar o modelo existente com dados de treino. Ao clicar nesse botão, através de um comando na consola DOS é corrido o programa/*script* elaborado em python com o nome "Harmonic_NN_CLASSIFIER4.1.py", onde corre a rede neuronal construída com base no conceito de *Multi-Layer Perceptron Classifier* (MLPC), apresentado na secção 3.3.5. O botão **START PROCESSING** corre o programa/*script* python também elaborado, "data_processing.py", responsável pelo pré-processamento de dados e pela gestão da atualização do modelo de aprendizagem do classificador. Os dois programas em python são abordados pormenorizadamente na secção seguinte.

4.4 Modelos de Machine Learning e resultados de aprendizagem

Como foi referido na secção 3.2, qualquer algoritmo de *machine learning* tem melhores resultados se os atributos do objeto de estudo (este caso os rolamentos) forem os que mais reflitam na diferenciação, e quanto maior for o número de atributos, mais precisa é a saída, mas o tempo de processamento aumenta drasticamente. As redes neuronais já existem há bastante tempo, e a sua formulação está bastante sólida, o maior problema reside sempre no poder computacional que um dado sistema complexo exige. Por esse motivo o número de variáveis de entrada deve ser o menor que possível, existindo até algoritmos só para avaliar se são efetivamente os melhores atributos que descrevem um dado sistema. Isto é, sem elaborar uma rede neuronal em que cada variável de entrada contribui para o resultado final, determinam a relação entre variáveis através de cálculos de distâncias como a euclidiana e Mahalanobis.

Normalmente o processo de aplicação de um algoritmo de *machine learning* segue os seguintes passos:

1. Seleção de atributos que contribuam na previsão;
2. Descriminar os resultados por classes para treino (relacionando os parâmetros "Nº de linhas de dados obtidos" e "Diagnóstico" da Tabela 4-1);
3. Treinar um modelo;
4. Calcular a performance do algoritmo em dados de teste.

De acordo com M. Bowles [11], a preparação e seleção de atributos significativos, estima-se que seja entre 80 a 90 porcento to tempo necessário para desenvolver um modelo de *machine learning*.

Os atributos que refletem o estado de um rolamento e que antecipam uma falha podem ir desde vibrações de ressonância da máquina, ruido, variações da velocidade e/ou carga, frequências dos constituintes dos rolamentos, temperatura, quantidade de massa lubrificante, horas de trabalho e até estatística das avarias que ocorreram na máquina (MTBF). Posto isto os indicadores ou atributos do sistema que foram usados para a previsão do tempo de vida dos rolamentos ou estendendo até ao tempo de vida da máquina, estão presentes na Tabela 4-4.

Tabela 4-4 - Dados de entrada para o processo de aprendizagem

BPFO_thd	Distorção harmónica total na "frequência" de passo dos elementos rolantes no anel exterior
BPFI_thd	Distorção harmónica total na "frequência" de passo dos elementos rolantes no anel interior

BSF_thd	Distorção harmónica total na “frequência” dos elementos rolantes
FTF_thd	Distorção harmónica total na “frequência” correspondente à velocidade
Fv_thd	Distorção harmónica total da “frequência” do veio do motor
Thd	Distorção harmónica total do sinal medido
distrib_freq	Média ponderada da frequência
Mean_temperature	Temperatura média

As distorções harmónicas foram calculadas com a equação (3.13). A “distorção harmónica total do sinal medido”, não tem o propósito de demonstrar a distorção face à frequência principal, até porque são incluídas todas as frequências lidas e não as harmónicas da frequência principal. O objetivo é ter um valor que caracterize o espectro no domínio da amplitude, sendo que para o domínio da frequência é calculada a frequência média através da equação da média ponderada (4.1) onde (F_i) é o valor da frequência do harmónico de índice (i) e (A_i) é a respetiva amplitude.

$$\bar{x} = \frac{\sum A_i \times F_i}{\sum A_i} \tag{4.1}$$

Definidos os atributos/indicadores segue-se o 3º passo onde se treina o modelo. Foi consultado um guia elaborado por *developers* do Scikit-learn (ver secção 3.3.5) com base no objetivo e no tamanho da base de dados como se pode ver na Figura 4-13.

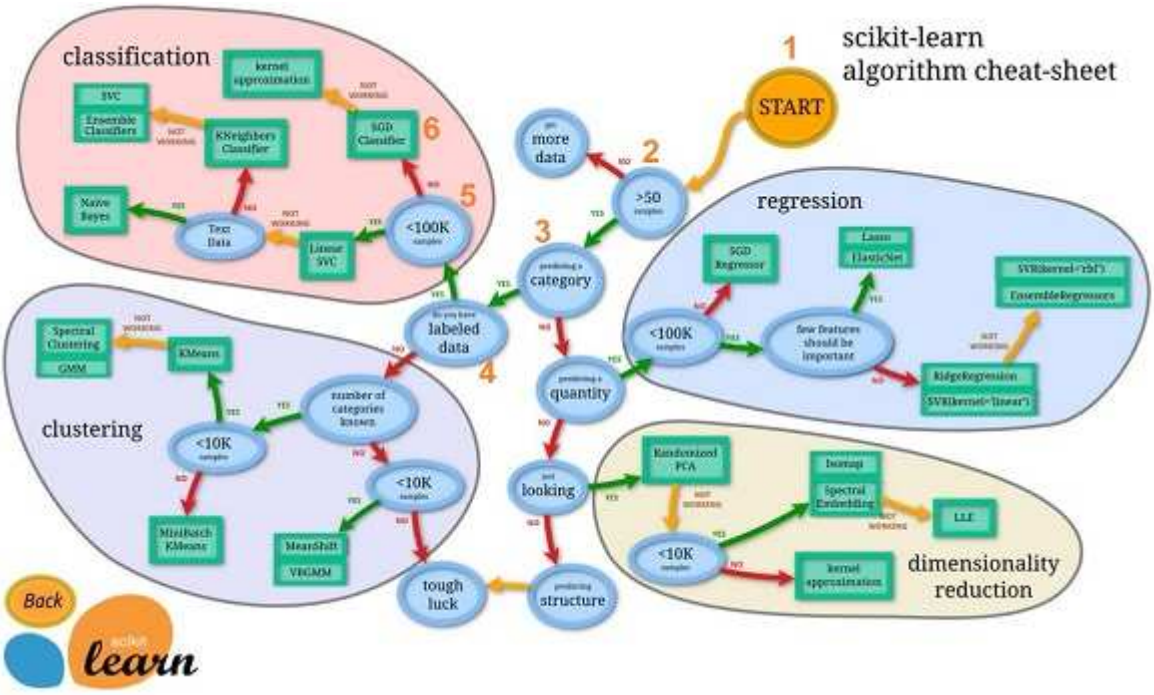


Figura 4-13 - Tipos de algoritmos em scikit-learn[18]

Seguindo os passos marcados na Figura 4-13, obtém-se a sugestão de usar um algoritmo baseado na Descida de Gradiente com aprendizagem estocástica. Este método conhecido, do inglês, por *Stochastic Gradient Descent* (SGD) é uma abordagem simples, mas muito eficiente para a aprendizagem de classificadores lineares sob funções de perda convexa (*convex loss funtions*), tais como Máquinas de Vetores de Suporte (secção 3.3.4) e Regressão Logística[41]. Embora o SGD tenha estado na comunidade de *machine learning* há bastante tempo, recebeu uma atenção considerável recentemente no contexto da aprendizagem em larga escala. Desta forma o método escolhido para fazer a aprendizagem do tempo de vida foi um classificador com uma rede neuronal multicamada(MLP) que pode ser encontrado no módulo do Scikit-learn e pode ser configurado segundo vários parâmetros, abordados na secção 3.3.5. Esses parâmetros são, por exemplo, o máximo número de iterações, a função de ativação, a "velocidade de aprendizagem", "validação", o momento e o número de "mini-grupos". A Figura 4-14 é um pequeno excerto do programa/script desenvolvido em python com o propósito de criar um modelo que descreva a classificação ou, de certa forma, determine do tempo de vida de um rolamento segundo um determinado espectro medido (Harmonic_NN_CLASSIFIER.py).

```
from sklearn.neural_network import MLPClassifier

modelo = MLPClassifier(activation='relu', alpha=0.00001, batch_size='auto', beta_1=0.9,
       beta_2=0.999, early_stopping=False, epsilon=1e-08, learning_rate='adaptive',
       learning_rate_init=0.001, momentum=0.9,hidden_layer_sizes=(13,9,5),max_iter=2000,
       nesterovs_momentum=True, power_t=0.5, random_state=None,
       shuffle=True, solver='lbfgs', tol=0.00001, validation_fraction=0.01,
       verbose=False, warm_start=False)
```

Figura 4-14 - Parâmetros do algoritmo de machine learning MLPC

O algoritmo para em duas situações, quando atinge o número máximo de iterações definido na Figura 4-14 como "max_iter=2000" e quando tem um erro inferior à tolerância definida na mesma figura por "tol=0,00001". Este pequeno excerto constrói e interrelaciona uma rede neuronal artificial com o aspeto da figura seguinte (Figura 4-15), que tal como mencionado na secção 3.3.5, segue uma metodologia de *Feed-forward* e uma determinação do erro por *Back-propagation*.

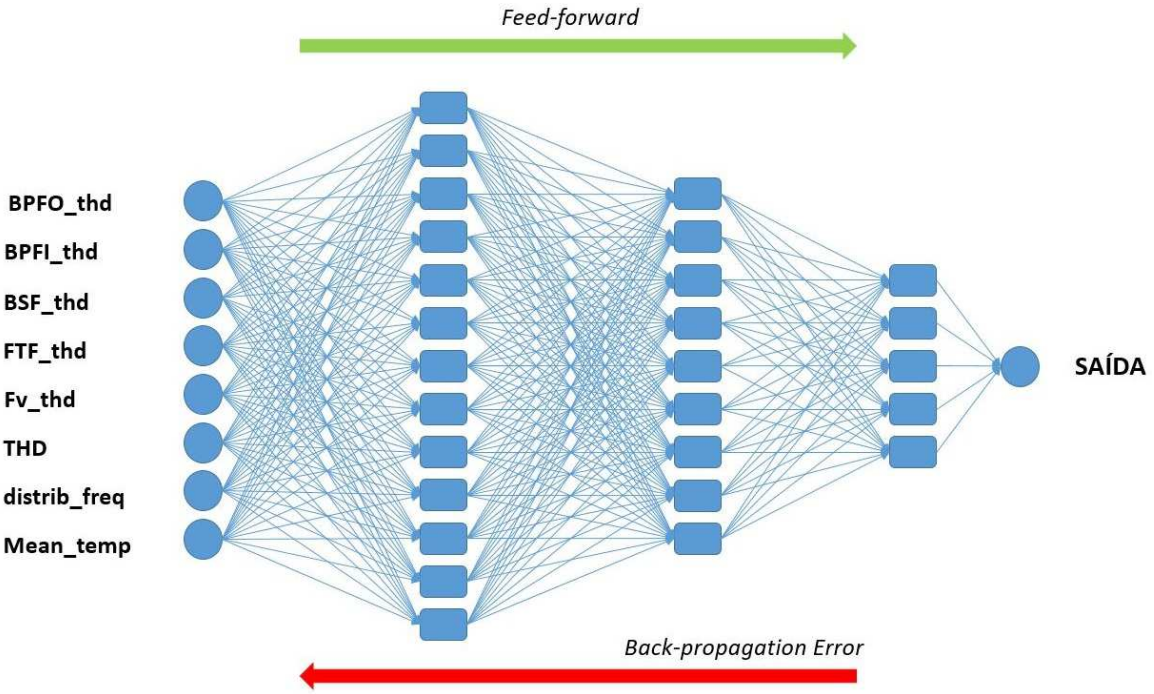


Figura 4-15 - Rede Neuronal Artificial Usada

O algoritmo preditivo não pode correr indefinidamente, isto até porque já foram mencionados os critérios de paragem. O fluxograma da figura seguinte (Figura 4-16) explica, de uma forma superficial, como é que o algoritmo volta a correr novamente, sendo que posteriormente no fluxograma da Figura 4-18 são apresentados os critérios para executar novamente o algoritmo.

A Figura 4-16 ilustra de uma forma superficial como são processados os dados até se obter a previsão do tempo de vida do equipamento. Este começa quando é pressionado o botão **START PROCESSING** mencionado na secção anterior. São lidas 5 linhas da base de dados de chegada do microcontrolador com valores de frequências, as respetivas amplitudes e a temperatura; sendo também lidos os parâmetros guardados no ficheiro de configuração. Depois do tratamento de dados, estes são submetidos ao modelo resultante do algoritmo preditivo determinando uma previsão do tempo de vida, sendo de seguida guardada uma linha, na tabela "bigdata", que corresponde no fluxograma à saída do processo "data_processing". Em diversas situações este script exige que seja calculado um novo modelo correndo "Harmonic_NN_CLASSIFIER.py". Isto permite uma atualização presumidamente de melhoria. Depois de atualizado o modelo, obtêm-se uma previsão que é guardada numa tabela da base de dados especificamente para ser usada como display no NodeRED.

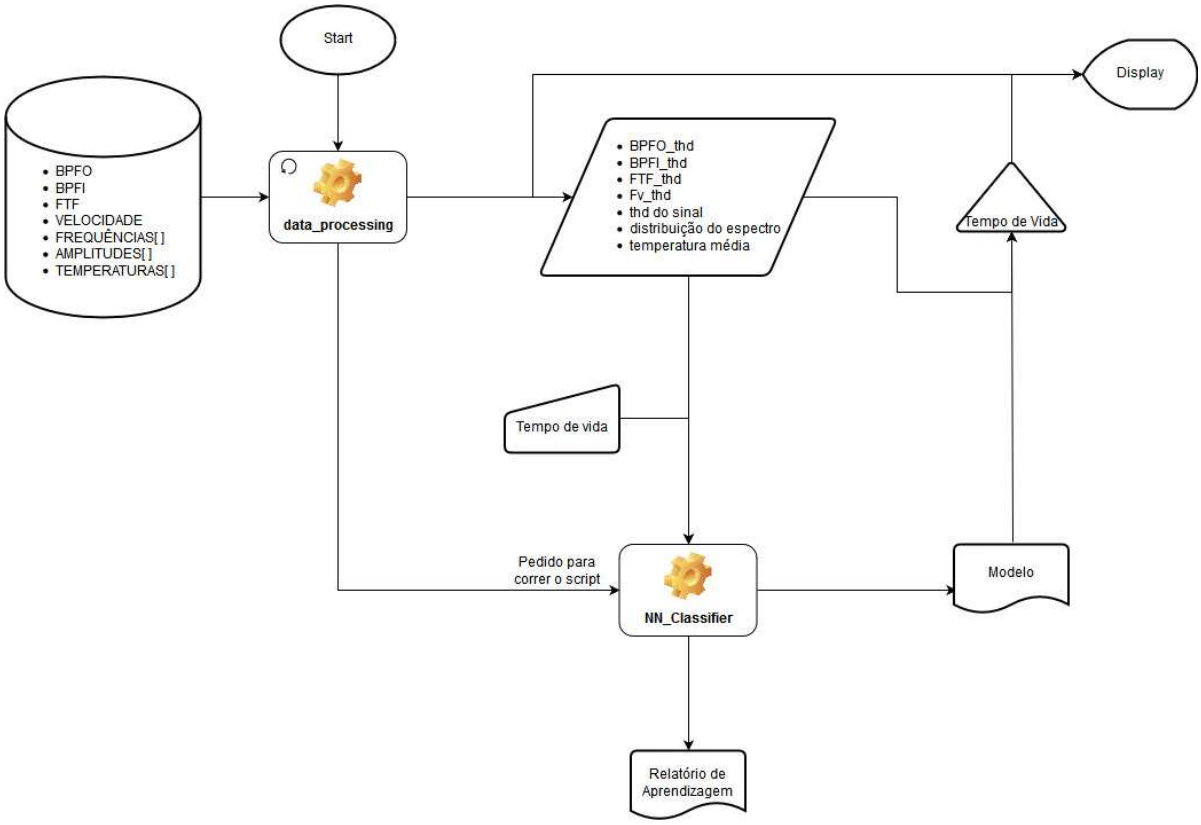


Figura 4-16 - Processo de aquisição, tratamento de dados e previsão do tempo de vida dos rolamentos

A legenda da simbologia utilizada na elaboração de todos os fluxogramas deste projeto segue a norma ISO 5807:1985 [42] e pode ser consultada no Anexo A.

Sempre que o script "Harmonic_NN_Classifier.py" corre, obtém-se um novo modelo que é guardado com o módulo/biblioteca "joblib" [43], e é apresentado, na consola DOS, um relatório da eficiência do modelo aprendido. A figura seguinte é um exemplo desses relatórios.

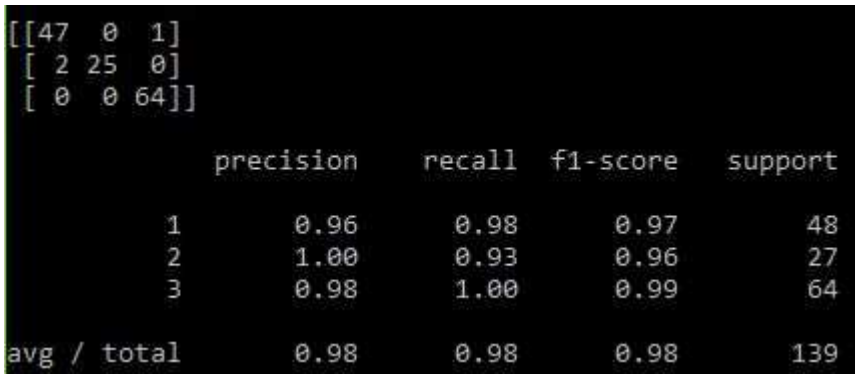


Figura 4-17 - Relatório do resultado da aprendizagem no DOS

A Figura 4-17 é constituída por uma matriz na parte superior e uma tabela na parte inferior. A tabela apresenta a percentagem de classificações corretas na coluna *recall*, sendo que as outras colunas são o cálculo da precisão (*precision*) e o *score* determinado pelo número de iterações que o programa teve de correr para conseguir um bom desempenho. A matriz é denominada de *confusion matrix* que pretende demonstrar como foram feitas todas as classificações, neste caso, de 139 linhas de teste em que apenas existiam 3 escolhas possíveis, Diag1, Diag2 e Diag3 como se pode ver na Tabela 4-5. Nesta tabela é feita a comparação entre o número de linhas de teste reais de um tipo e os assumidos pelo classificador. Por exemplo a previsão é de que existem 49 linhas (47+2+0) do Diag1, quando na realidade existem apenas 47.

Tabela 4-5 - Exemplo de matriz de confusão

		Previsão		
		Diag1	**Diag2**	**Diag3**
Realidade	**Diag1**	47	0	1
	Diag2	2	25	0
	Diag3	0	0	64

A ideia de manter uma aprendizagem contínua é de atualizar o modelo ao longo do tempo. Isto é, embora se consiga reunir inúmeros dados de diferentes estados de degradação de rolamentos/máquinas, obtendo um modelo com esta informação e usa-lo indefinidamente para classificar dados de entrada do sistema, não é uma aprendizagem contínua (o algoritmo corre uma

vez). É por isso que foram elaboradas várias condicionantes dentro do programa data_processing.py que iniciam o programa Harmonic_NN_Classifier.py, atualizando o modelo.

Este sistema está desenhado para avaliar o tempo de vida de rolamentos em semanas, podendo classifica-los desde zero a até um valor teoricamente infinito. Isto porque o sistema vai reconhecendo novos prazos de duração dos rolamentos à medida que o tempo passa. Segundo o fabricante, mediante a aplicação do rolamento, este terá um determinado tempo de vida, contudo devido a por exemplo, variações de carga causadas pelo processo e montagens erradas, este tempo de vida diminui.

No fluxograma da Figura 4-18 quando o sistema inicia, são definidos alguns parâmetros. Os contadores "previ_1" e previ_2", correspondendo à contagem de má classificação por defeito e à contagem de má classificação por excesso respetivamente, são inicializados a zero. A variável (X_i) guarda o valor da previsão em número de semanas. O contador do número de iterações do sistema (counter), responsável pelo controlo temporal, é também inicializado a zero. Cada linha de médias e distorções harmónicas calculadas, corresponde aproximadamente a 1 min, logo como se monitoriza o tempo de vida dos rolamentos, é necessário ter a noção do tempo que passa, mas em vez de usar o tempo real, é feita uma contagem de linhas processadas. Caso o sistema preveja que o rolamento tem o tempo de vida de 5 semanas e ao fim de uma semana, que é o mesmo que dizer ao fim de ≈10080 iterações, ainda continua a classificar da mesma forma, todas as linhas da semana imediatamente anterior devem ser reclassificadas para 6 semanas. Como é refeita uma reclassificação, é necessário correr novamente ao algoritmo MPLC para atualizar o modelo. Antes de chegar ao fim de uma semana, (n) iterações mais cedo (por defeito, n = 100), é calculada com Equação (4.2), a percentagem do quanto mudou a previsão face ao inicio da semana. Caso este valor seja superior a 9% então significa que o sistema já atualizou a classificação pela rotina secundária, caso contrário, então são reclassificadas todas as 10080 linhas anteriores para corresponderem a Xi + 1 semana.

$$\lambda = \left(1 - \frac{x_i \times n}{\sum_j^n previ_{n-j}}\right) \times 100\% \tag{4.2}$$

Onde (x_i) é a previsão no início do ciclo de uma semana, (n) é número de linhas anteriores ao fim de ciclo que a fórmula verifica se foram atualizadas e (j) é um contador de 0 a (n).

A rotina secundária acontece a cada 1000 iterações (≈16 horas). Esta rotina consiste no cálculo de percentagens onde,

- pup – percentagem de classificações em que a previsão é maior que a inicial (previ = Xi+1)
- pdown – percentagem de classificações em que a previsão é menor que a inicial (previ = Xi-1)

- pequal – percentagem de classificações em que a previsão é igual à inicial (previ = Xi)

A percentagem maior entre “pup” e “down” faz com que as últimas 1000 linhas sejam reclassificadas com a respetiva previsão, previ+1 ou -1 (aumentando ou diminuindo em uma semana) e é atualizado o modelo com o algoritmo MLPC. Se a “pequal” for a maior das percentagens, então não é atualizado o modelo e apenas são reinicializados os contadores e iterado o valor de “marker”.
Desta forma, com a sub-rotina a ocorrer a cada 16 h, o sistema ganha mais robustez contra eventos aleatórios de aumento/diminuição de carga na máquina, rotura de elementos constituintes do equipamento e aumentos pontuais de carga. Desta forma, o tempo de vida do rolamento é adaptado.

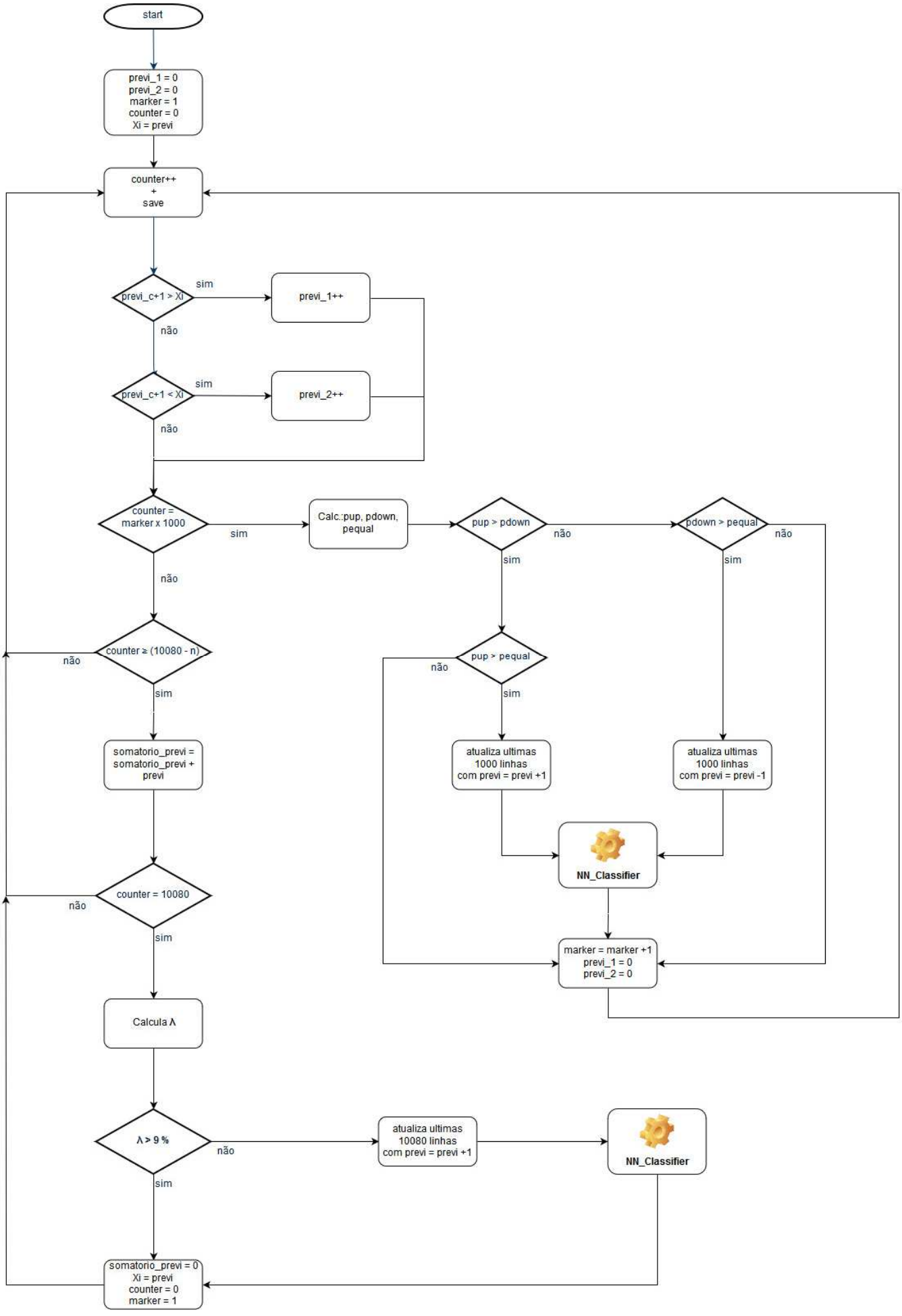

start
previ_1 = 0
previ_2 = 0
marker = 1
counter = 0
Xi = previ
counter++
+
save
previ_c+1 > Xi
sim
previ_1++
não
previ_c+1 < Xi
sim
previ_2++
não
counter =
marker x 1000
sim
Calc.:pup, pdown,
pequal
pup > pdown
não
sim
pdown > pequal
não
sim
não
counter ≥ (10080 - n)
não
sim
pup > pequal
não
sim
somatorio_previ =
somatorio_previ +
previ
atualiza ultimas
1000 linhas
com previ = previ +1
atualiza ultimas
1000 linhas
com previ = previ -1
NN_Classifier
counter = 10080
não
sim
marker = marker +1
previ_1 = 0
previ_2 = 0
Calcula λ
λ > 9 %
não
atualiza ultimas
10080 linhas
com previ = previ +1
NN_Classifier
sim
somatorio_previ = 0
Xi = previ
counter = 0
marker = 1

Figura 4-18 - Algoritmo proposto para Aprendizagem contínua

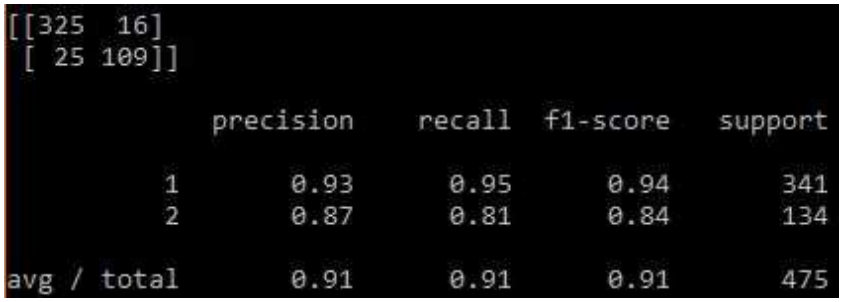
Como já foi referido, cada 5 linhas correspondem aproximadamente 1 minuto, por isso em termos temporais, foram recolhidos dados de aproximadamente, 22 horas de trabalho do motor do Diag1, 8,7 horas do motor do Diag2 e apenas 3,5 horas do Diag3.

O ideal seria acompanhar a vida toda de um rolamento/motor num determinado local de instalação, ao qual está conferida uma certa carga e periodicidade. Adicionalmente o espetro de início deveria ser tomado como referencia ao estado ótimo da máquina.
O sistema não teve tempo suficiente de se auto ajustar para fazer iterações na previsão do tempo de vida em semanas e por isso é que foram atribuídos os valores de 1, 2 e 3 à previsão de dados do Diag1, Diag2 e Diag3 respetivamente.
Após a recolha de dados do motor do Diag1, foi efetuada a aprendizagem do modelo por forma a verificar se a normalização estava a ser feita e se de um modo geral o programa corria sem quaisquer problemas, tendo resultado no relatório que se pode ver na Figura 4-19. Este resultado já era espectado visto tanto os dados da aprendizagem, como os de teste, eram do Diag1, então obteve-se 100% de precisão.

```
[[344]]
             precision    recall  f1-score   support

          1       1.00      1.00      1.00       344

avg / total       1.00      1.00      1.00       344
```

Figura 4-19 - Relatório da aprendizagem para 1 opção de saída

De seguida foram adicionadas, à tabela de aprendizagem, linhas do Diag2, previamente processadas por filtragem, cálculo de distorções harmónicas, médias, etc. Foi elaborado um novo modelo, cujo relatório que agora envolve a classificação de 2 tipos de motores, apresenta-se na figura seguinte.

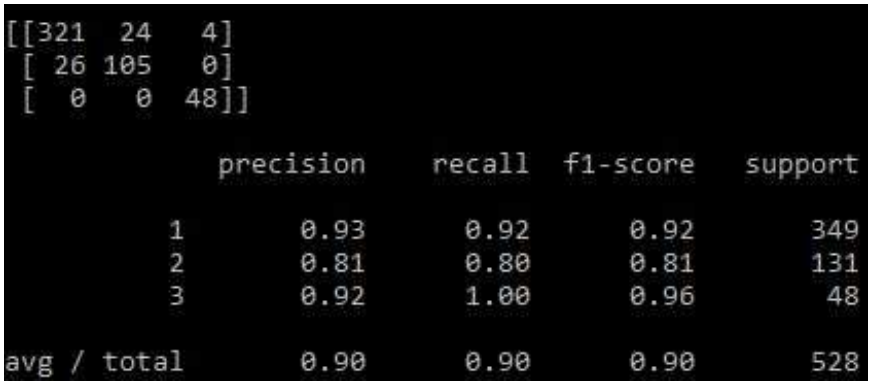


```
[[325  16]
 [ 25 109]]

             precision    recall  f1-score   support

          1       0.93      0.95      0.94       341
          2       0.87      0.81      0.84       134

avg / total       0.91      0.91      0.91       475
```

Figura 4-20 - Relatório da aprendizagem para 2 opções de saída

Depois desta aprendizagem verifica-se que no teste foram escolhidas aleatoriamente 475 linhas de dados, errando apenas em 25 que deveriam ser do Diag2 e foram classificadas como sendo do Diag1, e 16 linhas que deviam ser classificadas como Diag1 e forma classificadas como sendo do Diag2. Verifica-se também uma taxa de sucesso de 91%.

Por último foram utilizados os restantes dados, correspondentes ao Diag3 e foi elaborado um movo modelo que seja capaz de distinguir os 3 tipos que se pode ver na seguinte figura.



```
[[321  24   4]
 [ 26 105   0]
 [  0   0  48]]

             precision    recall  f1-score   support

          1       0.93      0.92      0.92       349
          2       0.81      0.80      0.81       131
          3       0.92      1.00      0.96        48

avg / total       0.90      0.90      0.90       528
```

Figura 4-21 Relatório da aprendizagem para 3 opções de saída

Na Figura 4-21 verifica-se uma taxa de sucesso em média de 90%, constatando-se também que ao adicionar uma máquina diferente ao sistema, esta ou pelo avançado estado de degradação ou pelas dimensões, destaca-se, visto que modelo de machine learning apenas confundiu 4 linhas do Diag1 pelo Diag3, mas não confundiu nenhuma linha do Diag3 tendo obtido 100% na sua classificação.

O algoritmo foi executado várias vezes visto que escolhe sempre aleatoriamente valores da base de dados, por forma a obter mais "tabelas de verdade" (*confusion matrix*) obtendo os resultados da figura seguinte.

Teste 2

```
[[321  17   2]
 [ 21 112   0]
 [  2   0  53]]
```

	precision	recall	f1-score	support
1	0.93	0.94	0.94	340
2	0.87	0.84	0.85	133
3	0.96	0.96	0.96	55
avg / total	0.92	0.92	0.92	528

Teste 3

```
[[316  31   0]
 [ 18 112   0]
 [  1   1  49]]
```

	precision	recall	f1-score	support
1	0.94	0.91	0.93	347
2	0.78	0.86	0.82	130
3	1.00	0.96	0.98	51
avg / total	0.91	0.90	0.90	528

Teste 4

```
[[316  22   4]
 [ 19 107   1]
 [  3   0  56]]
```

	precision	recall	f1-score	support
1	0.93	0.92	0.93	342
2	0.83	0.84	0.84	127
3	0.92	0.95	0.93	59
avg / total	0.91	0.91	0.91	528

Teste 5

```
[[321  17   2]
 [ 21 112   0]
 [  2   0  53]]
```

	precision	recall	f1-score	support
1	0.93	0.94	0.94	340
2	0.87	0.84	0.85	133
3	0.96	0.96	0.96	55
avg / total	0.92	0.92	0.92	528

Figura 4-22 - Relatórios de várias aprendizagens

Por forma a tornar os resultados um pouco mais percetíveis de interpretação segue-se a Figura 4-23.

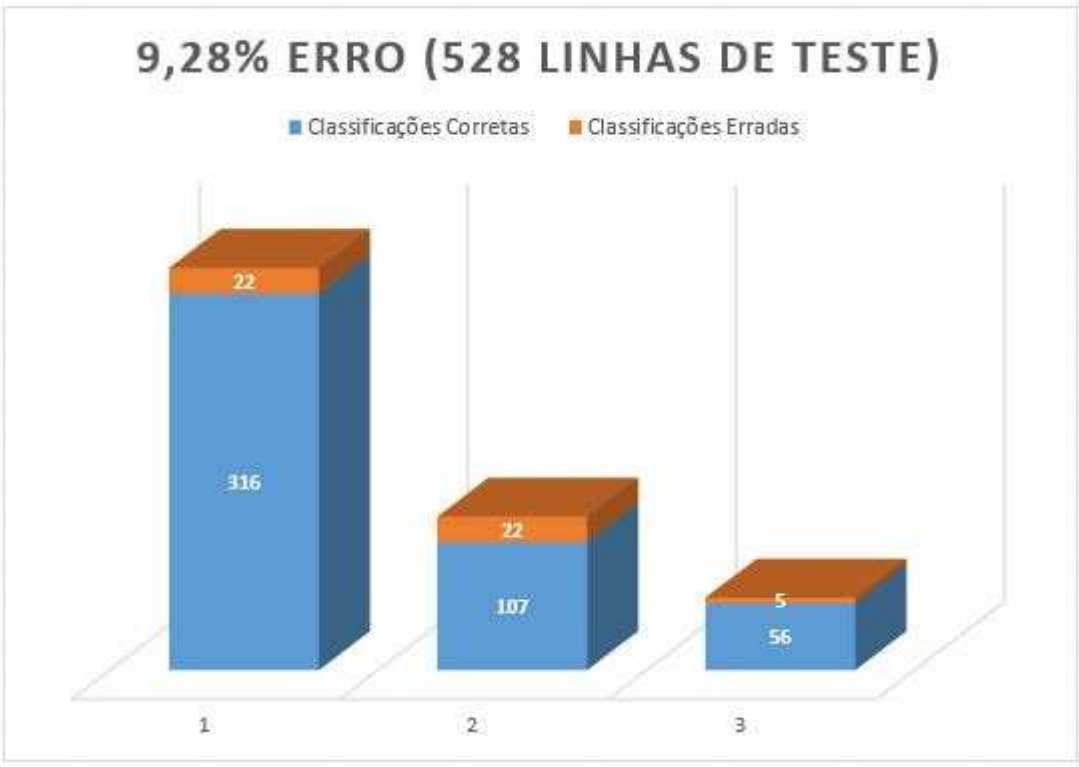


Figura 4-23 - Percentagem de Erro (528 linhas de teste)

O número casos do tipo 1, 2 e 3 deveria de ser próximo, contudo apesar da grande diferença entre o tipo 1 e os restantes, verifica-se uma estabilização das más classificações, isto é não é proporcional, o que significa que o modelo está realmente a funcionar.

5 Conclusões

A manutenção preditiva é o sonho de qualquer empresa que seja constituída por maquinaria submetida a esforços em laboração contínua, que cada vez mais, se torna numa realidade. Quando se torna possível prever a avaria de um dado equipamento, seja esta por métodos estatísticos ou por outro tipo de indicadores, é possível preparar paragens de secção atempadamente, reduzindo custos em paragens não planeadas que acarretam, por vezes, encomendas imediatas de peças, e por isso mais caras.

Este projeto teve como objetivo apresentar uma solução para prever o tempo de vida de um rolamento, que vai desde o hardware onde se projeta um sensor, que se pretende que seja de baixo custo, até ao software onde é testado um método preditivo, proposto, associado às redes neuronais artificiais. As soluções já existentes, são bastante dispendiosas e por vezes são vendidas como um modo de monitorização e não como uma forma de prever uma avaria.

O sistema de recolha de dados, mostrou-se funcional e permitiu a recolha de bastante informação.

O calculo das distorções harmónicas, médias e todos os outros indicadores usados no algoritmo, mostrou-se ser bastante útil, porque caso se usassem os dados vindos do microcontrolador diretamente, este obteria padrões de funcionamento da máquina e não do estado de degradação da mesma. O objetivo é ser transversal à dimensão e velocidade da máquina e a medição de vibração no domínio do tempo, está diretamente relacionada com estes dois parâmetros. Por este motivo é que a medição de vibração no domínio da frequência favorece na direção deste objetivo.

A taxa de sucesso na classificação com 2 ou 3 motores foram bastante altas tendo sempre atingindo valores entre os 83 e os 97%. Uma particularidade interessante foi quando o modelo apenas conhecia objetos de estudo 1 e 2, este penas podia fazer uma escolha binária em que 1 significava mau estado e 2 significava em bom estado, e quando foram submetidos dados do Diag3 a este modelo, este classificou 77% das vezes como sendo um motor 1, isto é que estava em mau estado o que estava correto.

Um problema encontrado em ambiente fabril foi a rede de WiFi, que mesmo sendo dedicada (local), por vezes perdia sinal devido a interferências eletromagnéticas e das redes existentes que impediam o funcionamento contínuo das rotinas de aprendizagem.

Seria interessante usar apenas soluções de hardware já existentes e usar esse como ponto de partida para o desenvolvimento de rotinas e teste de algoritmos de aprendizagem, visto que o

desenvolvimento do sensor demonstrou ser bastante desafiante e consumidor de tempo sem ter surtido quaisquer resultados.

A representação gráfica em web-browser (NodeRED) foi também um desafio porque é um conceito muito recente e não existe tanta informação como aquela que é necessária. Tem uma interface intuitiva, por ser gráfica, mas apresenta um grau de complexidade acrescida no que toca a configurações em JSON.

Como trabalho futuro segue-se a monitorização e aprendizagem de vários estados da máquina. O MLPC tem uma performance melhor para um elevado número de dados. Este projeto pode ser dividido em 3 projetos para aumentar a eficiência de cada uma destas vertentes, sendo estas, a projeção do sensor de baixo custo, a otimização das interfaces gráficas e a melhoria e desenvolvimento de algoritmos preditivos. No caso de seguir apenas a vertente dos algoritmos preditivos, uma alternativa ao desenvolvimento de hardware, é adquirir sensores IIoT dedicados para a industria.

A base de dados pode ser bastante otimizada, através da normalização das tabelas e do desenvolvimento de interfaces entre programas que não tenham de usar uma tabela da base de dados como meio de transmissão de valores.

A integração do sistema SAP é uma oportunidade de tirar partido das funcionalidades estatísticas desta plataforma, usando-as no algoritmo de *machine learning*, não sendo necessária a análise de um especialista com tanta frequência. Por este motivo, este tópico que ficou em falta é um ponto de partida importante para o trabalho futuro.

Referências

[1] T. N. P. Cacia, «Monografia da Fábrica de Cacia». Cacia, Aveiro, pp. 1–43.

[2] Yokogawa, «Pulp and Paper Aplications». [Em linha]. Disponível em: https://www.yokogawa.com/us/library/resources/application-notes/pulp-and-paper-applications/. [Acedido: 03-Nov-2017].

[3] Sergiusz Prokurat, «Industry 4.0. Are we there yet? | WBJ». [Em linha]. Disponível em: http://wbj.pl/industry-4-0-are-we-there-yet/.

[4] phpMyAdmin, «Bringing MySQL to the web», 2018. [Em linha]. Disponível em: https://www.phpmyadmin.net/. [Acedido: 04-Nov-2018].

[5] Oracle, «Top 10 Reasons to Choose MySQL for Next Generation Web Applications: A MySQL Whitepaper», n. August, p. 22, 2011.

[6] BSI, «EN 13306 Maintenance - Maintenance terminology», *BSI*, 2010. [Em linha]. Disponível em: http://irma-award.ir/wp-content/uploads/2017/08/BS-EN-13306-2010.pdf. [Acedido: 10-Fev-2018].

[7] Avi Nowitz, «The Economics of the Smart Factory How does Machine Learning Lower the Cost of Asset Maintenance», *May 24*, 2017. [Em linha]. Disponível em: https://www.presenso.com/single-post/2017/05/24/the-economics-of-the-smart-factory-how-does-machine-learning-lower-the-cost-of-asset-maintenance-part-1/. [Acedido: 19-Dez-2018].

[8] K. Krzyk, «Coding Deep Learning For Beginners — Types of Machine Learning», *Towards data science*. [Em linha]. Disponível em: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d. [Acedido: 10-Jun-2018].

[9] F. Tseng, «Notes on Artificial Intelligence», pp. 1–673, 2016.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 53, n. 9. 2013.

[11] M. Bowles, *Machine Learning in Python: Essential Techniques for Predictive Analysis*, vol. 1542. Indianapolis, Indiana, Canada: John Wiley & Sons, Inc., 2015.

[12] E. S. Yudkowsky, «An Intuitive Explanation of Bayes' Theorem», *An Intuitive Explanation of Bayesian Reasoning*, 2006. [Em linha]. Disponível em: http://yudkowsky.net/rational/bayes.

[13] A. Mammone, M. Turchi, e N. Cristianini, «Support Vector Machines», *Adv. Rev.*, vol. 1, n. 3, pp. 283–289, 2009.

[14] M. Kirk, *Thoughtful Machine Learning with Python*, First Edit. Sebastopol,USA: O'Reilly Media, Inc., 2017.

[15] E. Costa e S. Anabela, «Redes Neuronais», em *Inteligência Artifical Fundamentos e Aplicações*, Lindel., Lisboa, 2008, p. 12/607.

[16] Venelin Valkiv, «Creating a Neural Network from Scratch—TensorFlow for Hackers (Part IV)», 2017. [Em linha]. Disponível em: https://medium.com/@curiousily/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8.

[17] Jose Portilla, «A Beginner's Guide to Neural Networks in Python and SciKit Learn 0.18 | Springboard Blog», 2017. [Em linha]. Disponível em: https://www.springboard.com/blog/beginners-python-scikit-learn-0-18/.

[18] S. Developers, «Neural network models (supervised)», 2018. [Em linha]. Disponível em: http://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron.

[19] H.-C. Lin, Y.-C. Ye, B.-J. Huang, e J.-L. Su, «Bearing vibration detection and analysis using enhanced fast Fourier transform algorithm», *Adv. Mech. Eng.*, n. 10, p. 14, Out. 2016.

[20] BSIC, «Using Fourier transform to detect large orders: Part 1», 2016. [Em linha]. Disponível em: http://www.bsic.it/using-fourier-transform-detect-large-orders-part-1/.

[21] D. Felten, «Understanding Bearing Vibration Frequencies», *Easa Curr.*, n. September, pp. 1–3, 2003.

[22] Wikipedia, «Total harmonic distortion». [Em linha]. Disponível em: https://en.wikipedia.org/wiki/Total_harmonic_distortion.

[23] SKF, «SKF - Componentes e materiais». [Em linha]. Disponível em: http://www.skf.com/pt/products/bearings-units-housings/principles/general-bearing-knowledge/bearing-basics/components-and-materials/index.html. [Acedido: 14-Ago-2018].

[24] B. P. Graney e K. Starry, «Rolling element bearing analysis crosses threshold», *Mater. Eval.*, vol. 70, n. 1, pp. 78–85, 2011.

[25] Steve Hanly, «6 Ways to Measure Vibration», 2016. [Em linha]. Disponível em: https://blog.mide.com/6-ways-to-measure-vibrations. [Acedido: 29-Set-2018].

[26] Texas Instruments, «IoT made easy», 2015. [Em linha]. Disponível em: http://www.ti.com/ww/en/wireless_connectivity/sensortag/?INTC=SensorTag&HQS=sensortag. [Acedido: 29-Set-2018].

[27] JIB, «Influência da temperatura em rolamentos».

[28] IBM, «IBM Maximo APM - Predictive Maintenance Insights - Pricing - Portugal». [Em linha]. Disponível em: https://www.ibm.com/pt-en/marketplace/predictive-maintenance-insights/purchase. [Acedido: 27-Dez-2018].

[29] B. Earl, «ADS1115 Adafruit 4-Channel ADC Breakouts». Adafruit Industries, p. 21, 2017.

[30] Texas Instruments, «LM35 Precision Centigrade Temperature Sensors», n. November, pp. 1–13, 2013.

[31] T. components Incorporated, «LM7833/LM7847 1A Positive Voltage Regulator», 2007.

[32] T. Instruments, «Ads1113 Ads1114 Ads1115», *Datasheet*, 2009.

[33] Electrolude, «ER1122: Epoxy resin», *HK Wentworth Ltd*. pp. 1–3, 2015.

[34] General Electric Company, «190501 Velomitor CR Velocity Transducer Description», *Bently Nevada, Inc.*, n. 141636. Bently Nevada, Inc., Minden, Nevada, pp. 1–11.

[35] S. Winder, *Analog and Digital Filter Design*, 2nd ed., vol. 39, n. 5. Woburn, 2002.

[36] P. Technology, «PicoScope® 2000 Series», 2014. [Em linha]. Disponível em: https://www.picotech.com/oscilloscope/2000/picoscope-2000-specifications.

[37] Apache Friends, «XAMPP Apache + MariaDB + PHP + Perl», 2018. [Em linha]. Disponível em: https://www.apachefriends.org/index.html.

[38] N. Bearings, «Formulas to Calculate Bearing Frequencies», n. 1, pp. 1–61.

[39] JS Foundation, «Node-RED». [Em linha]. Disponível em: https://nodered.org/. [Acedido: 15-Jul-2018].

[40] Microsoft, «Visual Studio 2017». [Em linha]. Disponível em: https://visualstudio.microsoft.com/pt-br/downloads/?rr=https%3A%2F%2Fwww.google.com%2F. [Acedido: 29-Out-2018].

[41] W. Xu, «Towards Optimal One Pass Large Scale Learning with Averaged Stochastic Gradient Descent», Jul. 2011.

[42] International Organization for Standardization, «ISO 5807:1985 - Information processing -- Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts», 2005. [Em linha]. Disponível em: https://www.iso.org/standar

[43] Python Software Foundation, «joblib · PyPI», 2018. [Em linha]. Disponível em:

[44] E. Singer, «In Brain's 'Rich Club,' Meetings of the Mind | Quanta Magazine», *Quanta Magazine*, 2013. [Em linha]. Disponível em: https://www.quantamagazine.org/in-brains-rich-club-meetings-of-the-mind-20131024/.

Anexo I Circuito Elétrico

O esquema elétrico da Ilustração 1 foi desenvolvido com o software ORCAD. Um dos objetivos seria a elaboração de uma placa de circuito impresso com dimensões bastante reduzidas. Isto justifica o facto de terem sido selecionados tipos de encapsulamento como o 0603 e 0805 para os componentes eletrónicos. As referencias 0603 e 0805 são as dimensões da caixa dos componentes eletrónicos em polegadas, 0603 é na verdade a dimensão de 0,06''x0,03'' que em mm resulta em 1,6x0,8 mm.
O circuito centra-se à base da medição das grandezas físicas, sendo estas a vibração e a temperatura. A vibração, ou melhor dizendo, a aceleração dinâmica é obtida através do acelerómetro de 3 eixos ADXL335[29], e a temperatura através do sensor de temperatura LM35[30].

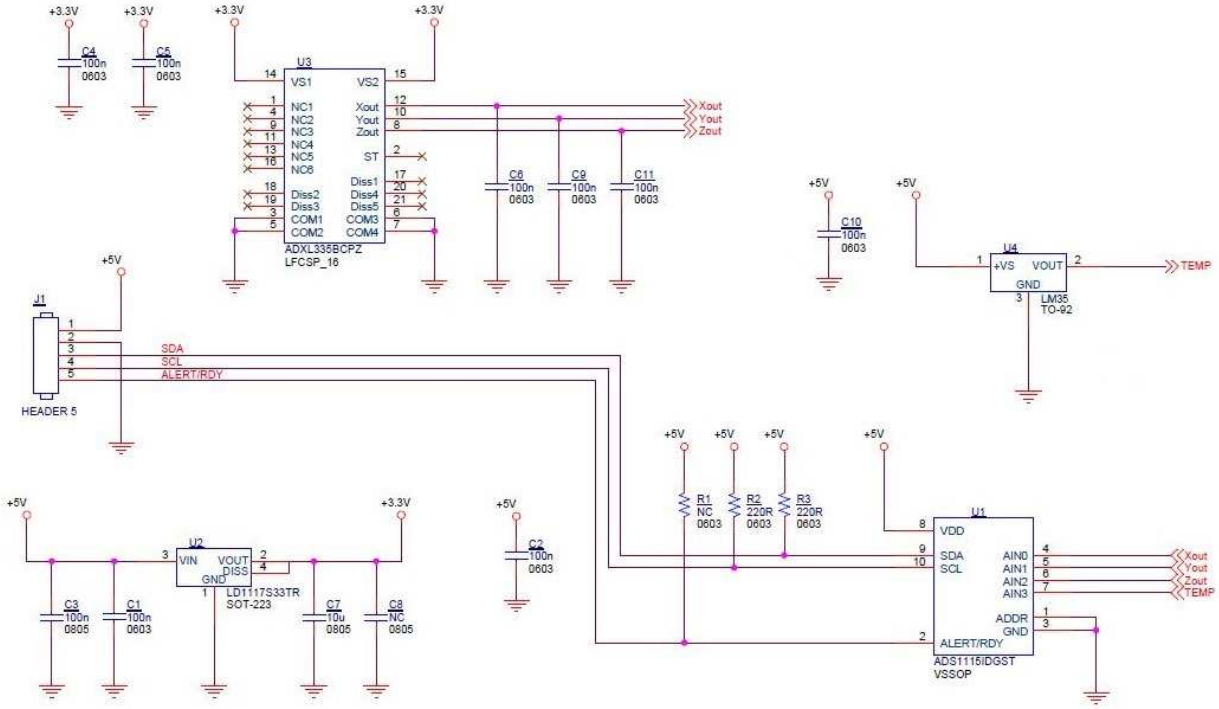


Ilustração 1 - Esquema elétrico do sensor

O circuito é alimentado a 5 V DC sendo que possui um regulador de tensão de 3V3 [31] para alimentar o acelerómetro e expansor de entradas analógicas (ADS1115). O ADS1115 [32] é um circuito integrado com 4 entradas analógicas, ficando uma para cada eixo do acelerómetro incluindo a leitura da temperatura, comunicando depois através do protocolo de comunicação I^2C. Todos os

condensadores e resistências são as recomendadas pelos circuitos típicos apresentados nas *datasheets* de cada circuito integrado.

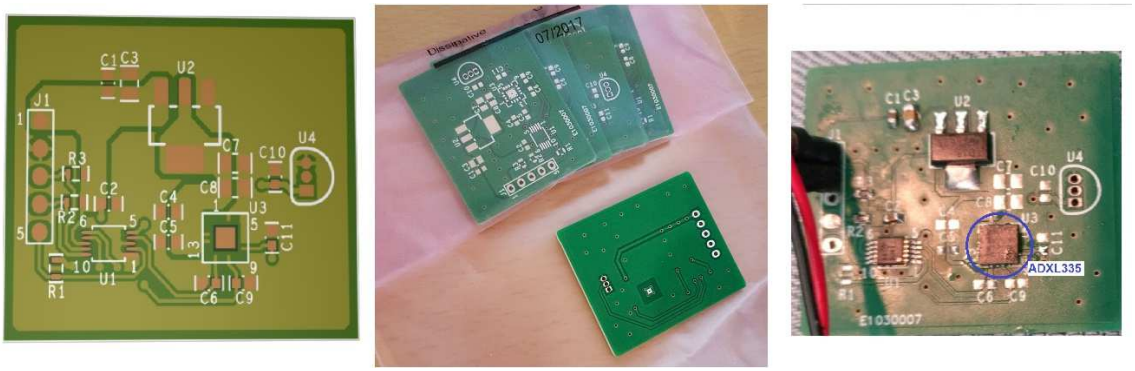


Ilustração 2 - PCB, à esquerda virtual GERBER VIEW, no centro sem componentes e à direita com alguns componentes

A Ilustração 2 demonstra o esquema elétrico da Ilustração 1 sob a forma de placa de circuito impresso, numa fase virtual (formato GERBER) do lado direito e do lado esquerdo, a placa de circuito impresso já elaborada.

O processo de construção do sensor não foi concluído porque a soldadura de componentes SMD não o permitiu. Foram soldados alguns componentes como como foi o caso do regulador de tensão e o módulo de 4 ADC's, contudo o acelerómetro é quase impossível de soldar convenientemente com uma estação de soldadura normal. Para este tipo de componentes é usado o método de *reflow* onde é aplicada uma pasta (mistura de solda com fluxo) nas *pads* do PCB, de seguida são colocados os componentes com uma máquina de *pick-and-place* seguindo para um forno de indução ou por infravermelhos onde a solda derrete soldando os componentes.

Anexo II Desenho mecânico

De forma similar aos sensores de vibração mais frequentes na industria, é necessária a proteção da placa de circuito impresso. Dessa forma, todo o circuito fica submerso numa resina "não condutora" ER1122 [33], dentro de um involucro, por forma a proteger hermeticamente o circuito e também para eliminar medidas de vibrações que não fossem provenientes do exterior da base do involucro.
Tendo como base o aspeto comum dos sensores de vibração tais como o 190501 Velomitor da Bently Nevada Inc. [34], o involucro foi desenhado com software CAD nomeadamente o Solidworks 2016. A idealização passou por algumas iterações sendo que o problema era encontrar uma forma do sensor de temperatura receber melhor a condução térmica. Para este feito, existe um furo de 2 mm de profundidade e 3 mm de diâmetro no centro do involucro, onde se pretende alojar o LM35, tal como se pode ver na Ilustração 3.

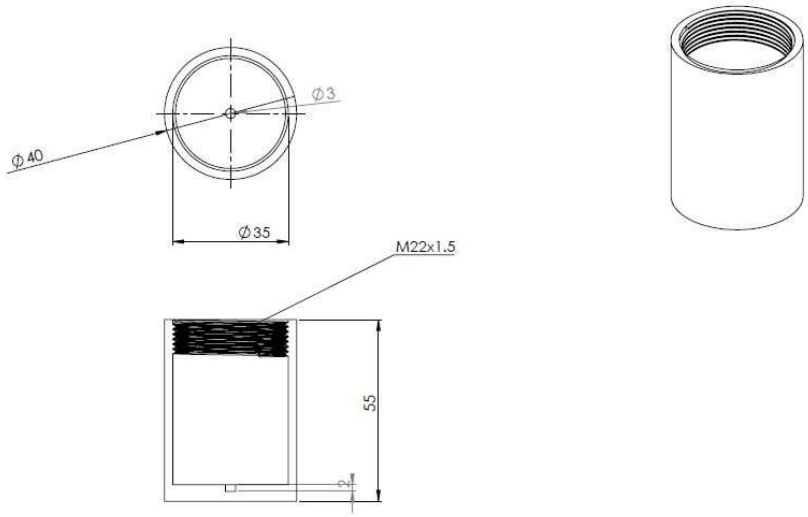


Ilustração 3 - Involucro do sensor (Solidworks 2016)

A tampa, na figura seguinte (Ilustração 4), era hexagonal com a semelhança de uma porca que roscava externamente no invólucro. Contudo foi adotada uma forma cilíndrica com o diâmetro do exterior igual ao do invólucro para lhe conferir uma estética moderna. Esta tampa possui na mesma dois entalhes para possibilitar uma fácil abertura/fecho.

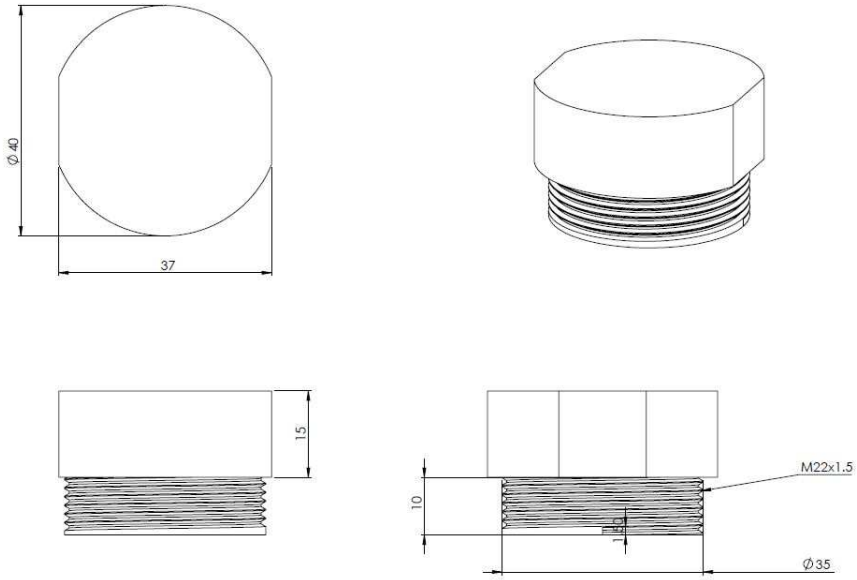


Ilustração 4 - Tampa do involucro do sensor (Solidworks 2016)

Na primeira iteração do desenho do involucro, foi elaborado um protótipo com uma impressora 3D que se pode ver na Ilustração 5, onde a rosca era exterior e não interior como no desenho final. Os componentes junto do protótipo foram usados numa montagem em *bread board* para testes do circuito e programação do microcontrolador.

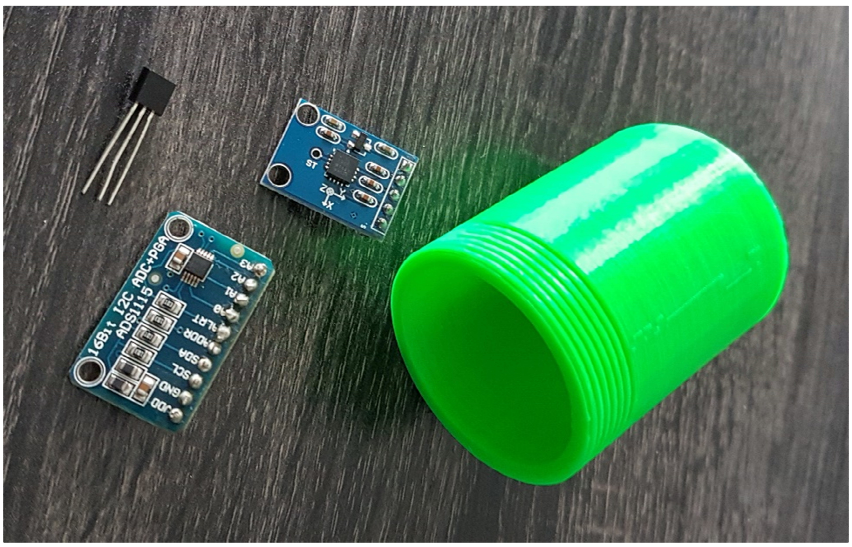


Ilustração 5 - Protótipo do invólucro do sensor

Não foram elaborados mais protótipos para além do apresentado na imagem anterior, porque como já foi referido (secção Anexo I) o sensor não foi concluído e não se justificava o avanço nesta vertente do projeto.