IPCA INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

SISTEMA TINYML PARA PREDIÇÃO DE ANOMALIAS EM MOTORES

TINYML SYSTEM FOR PREDICTIVE MOTOR ANOMALY

João Pedro da Costa Ferreira

Orientador

Doutor António Herculano Jesus Moreira

Dissertação apresentada

ao Instituto Politécnico do Cávado e do Ave

para obtenção do Grau de Mestre em Engenharia Eletrónica e de Computadores

dezembro, 2022

IPCA INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

SISTEMA TINYML PARA PREDIÇÃO DE ANOMALIAS EM MOTORES

TINYML SYSTEM FOR PREDICTIVE MOTOR ANOMALY

João Pedro da Costa Ferreira

Orientador

Doutor António Herculano Jesus Moreira

Dissertação apresentada

ao Instituto Politécnico do Cávado e do Ave

para obtenção do Grau de Mestre em Engenharia Eletrónica e de Computadores

dezembro, 2022

DECLARAÇÃO

Nome: João Pedro da Costa Ferreira
Endereço eletrónico: a13291@alunos.ipca.pt
Título da Dissertação: SISTEMA TINYML PARA PREDIÇÃO DE ANOMALIAS EM MOTORES
Orientador: Dr. António Herculano Jesus Moreira
Ano de conclusão: 2022
Designação do Curso de Mestrado: Mestrado em Engenharia Eletrónica e de Computadores

Nos exemplares das Dissertações /Projetos/ Relatórios de Estágio de mestrado ou de outros trabalhos entregues para prestação de Provas Públicas, e dos quais é obrigatoriamente enviado exemplares para depósito legal, deve constar uma das seguintes declarações:

☐ É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

☒ É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA DISSERTAÇÃO/TRABALHO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

☐ DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO/TRABALHO

Instituto Politécnico do Cávado e do Ave,22/12/2022

Assinatura: _____ Assinado por: **João Pedro da Costa Ferreira**
Num. de Identificação: 14077542
Data: 2022.12.22 23:55:51+00'00'

SISTEMA TINYML PARA PREDIÇÃO DE ANOMALIAS EM MOTORES

RESUMO

A manutenção preditiva em termos industriais tem se tornado num assunto de elevada relevância na Indústria 4.0, mais específico na área da *Internet of Things* (IoT), o mesmo se verifica com o uso de técnicas de *energy harvesting* em diversas áreas para sistemas de baixa energia. No entanto o processo de predição/deteção anomalia obriga a processar uma enorme quantidade de amostras de diferentes sensores para analisar o funcionamento de um equipamento e dessa forma esse processo fica dependente de sistemas de processamento de alta performance para a implementação de técnicas de *machine learning*.

Com o plano proposto neste projeto, pretende-se desenvolver um sistema embebido de manutenção preditiva, utilizando a placa de desenvolvimento ESP32 para a leitura de parâmetros associados a um motor industrial através de sensores externos, como por exemplo, acelerómetro e magnetómetro, em conjunto com a *framework Tensorflow Lite* pretende-se implementar um sistema preditivo de análise do modo/estado de funcionamento. Por fim, de forma a prolongar o tempo de vida do sistema, pretende-se estudar e simular o aproveitamento de energia através de fontes de energia existentes, sabendo-se que as energias remanescentes nestes equipamentos são pequenas e limitadas.

Ao acoplar o motor industrial juntamente com o sistema de inteligência artificial baseada em *supervised learning*, o mesmo é capaz de retornar valores de validação promissores, com uma precisão de 97,26%, mas com limitações associadas ao seu uso. Os resultados obtidos do aproveitamento de energia ao utilizar o *hardware* do sistema criado ficaram um pouco aquém das expectativas no que diz respeito ao tempo de vida útil final, no entanto a simulação efetuada com um *hardware* mais atualizado e eficiente em termos de *deep sleep* mostraram-se surpreendentes.

Palavras-chave: Edge AI, Aproveitamento de energia, Predição de anomalia, AI em manutenção, Indústria 4.0.

TINYML SYSTEM FOR PREDICTIVE MOTOR ANOMALY

ABSTRACT

Predictive maintenance in industrial terms has become a subject of high relevance in Industry 4.0, more specifically around the area of the Internet of Things (IoT), the same is true with the use of energy harvesting techniques in several areas for low energy systems. However, the anomaly prediction/detection process requires processing a substantial number of samples from different sensors to analyse the operation of an equipment and thus this process is dependent on high performance processing systems for the implementation of machine learning techniques.

With the proposed plan, in this project we intend to develop an embedded system of predictive maintenance, using the ESP32 development board to read parameters associated with an industrial engine through external sensors, such as accelerometer and magnetometer, together with the Tensorflow Lite framework is intended to implement a predictive system for analysis the mode/state of operation. Finally, in order to extend the life of the system, it is intended to study and simulate the use of energy through existing energy sources, knowing that the remaining energies in this equipment are small and limited.

When coupling the industrial engine with the artificial intelligence system based on supervised learning, it is capable to return promissory validity values, with an accuracy of 97.26%, but with limitations associated with their use. The results obtained from the use of energy when using the system hardware created fell slightly short of expectations about the final useful lifetime, however the simulation performed with a more updated and efficient hardware in terms of deep sleep proved surprising.

Keywords: Edge Ai, Energy Harvesting, Anomaly Prediction, Ai in Maintenance, Industry 4.0.

AGRADECIMENTOS

Aproveito desde já, com a finalização da dissertação, para agradecer gestos de pessoas que ao longo deste percurso deixaram a sua marca na minha vida, que estiveram presentes e ajudaram de uma forma ou de outra a tornar este percurso possível.

Em primeiro lugar um enorme obrigado aos meus pais, pois sem eles nada disto seria possível. Um obrigado pelo apoio incondicional, pela motivação, compreensão e um grande obrigado pelo incentivo fornecido diariamente. A toda a minha família e a minha namorada pela preocupação, pela paciência, pelo apoio diário e pelas palavras certas de motivação nos momentos mais difíceis.

Em segundo lugar gostaria de agradecer ao orientador professor António Moreira, pela sua disponibilidade no decorrer do projeto, pelo seu apoio, pelos sábios conselhos e pela partilha de conhecimento tornado possível a realização do mesmo. Ao professor João Vilaça por me acolher nas instalações do 2Ai e pela oportunidade de adquirir novas competências e de enriquecer a um nível profissional e pessoal.

Por último, a todos os meus companheiros e amigos um obrigado pela memorias criadas, pelos momentos de ajuda e pelas palavras de motivação nos momentos exatos, pois melhor do que ninguém sabem a dificuldade desta caminhada, em especial para o Claudino Costa, Pedro Lobo, Sérgio Pereira, André Giesteira, Gonçalo Pereira, Ricardo Rodrigues, Pedro Martins e Vítor Oliveira.

LISTA DE ABREVIATURAS E/OU SIGLAS

2Ai – *Applied Artificial Intelligence Laboratory*

2D – Duas dimensões

3D – Três dimensões

ADiT – *Applied Digital Transformation Laboratory*

AI – *Artificial Intelligence*

ANN – *Artificial Neural Network*

CeDRI – Centro de Investigação em Digitalização e Robótica Inteligente

CNN – *Convolution Neural Network*

COAP – *Constrained Application Protocol*

CSV – *Comma-separated values*

DL – *Deep Learning*

FFNN – *Feed Forward Neural Networks*

FFT – *Fast Fourier Transform*

HTTP – *Hyper Text Transfer Protocol*

I2C – *Inter-Integrated Circuit*

IC – *Integrated Circuit*

IoT – *Internet of Things*

LED – *Light-Emitting Diode*

ML – *Machine Learning*

MLP – *Multilayer-Perceptron*

MME – *Magneto-Mechano-Electric*

MQTT – *Message Queuing Telemetry Transport*

NB – *Naïve Bayes*

NN – *Neural Networks*

PCA – *Principal Component Analysis*

PCB – *Printed Circuit Board*

PdM – *Predictive Maintenance*

PLC – *Programmable Logic Controllers*

RL – *Reinforcement Learning*

RNN – *Recurrent Neural Network*

SAR – *Symposium on Applied Research*

SASYR – *Symposium of Applied Science for Young Researchers*

SPI – *Serial Peripheral Interface*

STFT – *Short Time Fourier Transform*

SVM – *Support Vector Machine*

TCP/IP – *Transmission Control Protocol/Internet Protocol*

TinyML – *Tiny Machine Learning*

WT – *Wavelet Transform*

LISTA DE SÍMBOLOS

A – Ampere
Ah – Ampere-hora
H – Horas
Iout – Corrente saída
J – Joule
K – Kelvins
mA – milliampere
mAh – miliampere-hora
mm – milímetro
V – Tensão
Vin – Tensão entrada
W – Watt
Wh – Watt hora
Ws – Watt segundo

ÍNDICE

RESUMO I
ABSTRACT III
AGRADECIMENTOS V
LISTA DE ABREVIATURAS E/OU SIGLAS VII
LISTA DE SÍMBOLOS IX
ÍNDICE XI
ÍNDICE DE FIGURAS XIII
ÍNDICE DE TABELAS XV
1. INTRODUÇÃO 1
1.1. ENQUADRAMENTO 2
1.2. IDENTIFICAÇÃO DO PROBLEMA 3
1.3. OBJETIVOS 4
1.4. DISSEMINAÇÃO DO TRABALHO 4
1.5. ESTRUTURA DO DOCUMENTO 5
2. ESTADO DA ARTE 7
2.1. TÉCNICAS DE *MACHINE LEARNING* 7
2.1.1. TÉCNICAS DE *SUPERVISED LEARNING* 8
2.1.2. TÉCNICAS DE *UNSUPERVISED LEARNING* 9
2.1.3. TÉCNICA DE *REINFORCEMENT LEARNING* 10
2.1.4. TÉCNICAS DE *NEURAL NETWORKS* 11
2.2. *TINY MACHINE LEARNING* 13
2.2.1. INTRODUÇÃO AO *TINYML* 13
2.2.2. BENEFÍCIOS DO *TINYML* 14
2.3. DETEÇÃO DE ANOMALIAS EM EQUIPAMENTOS INDÚSTRIAS 15
2.4. *ENERGY HARVESTING* 16
2.4.1. FONTES DE *ENERGY HARVESTING* 17
3. DESENVOLVIMENTO 19
3.1. FERRAMENTAS DE *HARDWARE* 19
3.2. AQUISIÇÃO E ALGORITMO DE CLASSIFICAÇÃO 22
3.2.1. CRIAÇÃO DO *DATASET* 22
3.2.2. DESENVOLVIMENTO DA REDE NEURONAL 26
3.3. SISTEMA DE MONITORIZAÇÃO E VISUALIZAÇÃO 28
3.3.1. PROTOCOLO DE COMUNICAÇÃO MQTT 30
3.3.2. FERRAMENTA DE CONEXÃO NODE-RED 30
3.3.3. BASE DE DADOS *INFLUXDB* E FERRAMENTA DE VISUALIZAÇÃO *GRAFANA* 31
3.4. ESTUDO E IMPLEMENTAÇÃO DA TÉCNICA DE *ENERGY HARVESTING* 32
4. TESTES E RESULTADOS 35
4.1. ENSAIO RELACIONADO COM A DETEÇÃO DE ANOMALIAS 35
4.2. ENSAIO DA PROPOSTA DE *ENERGY HARVESTING* 45
4.2.1. ESTUDO DO LOCAL DE AQUISIÇÃO. 45

4.2.2. CÁLCULO DA ENERGIA E DO TEMPO ÚTIL DA BATERIA 51
5. CONCLUSÕES E TRABALHOS FUTUROS 63
5.1. CONCLUSÕES 63
5.2. TRABALHOS FUTUROS 64
REFERÊNCIAS BIBLIOGRÁFICAS 65
ANEXOS 69

ÍNDICE DE FIGURAS

Figura 1.1 – Revoluções Industriais [1]. 1
Figura 1.2 – Esquema de implementação. a) Transmissão em tempo real de todos os sistemas; b) Análise com *tinyML* local e transmissão do estado apenas quando necessário 3
Figura 1.3 – Esquema o sistema proposto. 4
Figura 2.1 – Exemplo de algoritmos ML [9]. 7
Figura 2.2 – Estrutura da *decision tree* [13]. 8
Figura 2.3 – Exemplo de hiperplano em *Support Vector Machine* [14]. 9
Figura 2.4 – Exemplo de algoritmo *K-Means Clustering* [18]. 10
Figura 2.5 – Diagrama básico de *reinforcement learning* [20]. 11
Figura 2.6 – Exemplo de NN [22]. 11
Figura 2.7 – Exemplo de Convolução. a) 1º Passo b) 2º Passo c) Resultado final [26]. 12
Figura 2.8 – Tipos de *pooling* [26] 13
Figura 2.9 – *Machine learning* em sistemas embebidos [30]. 14
Figura 2.10 – Esquema de captação de energia magnética [42]. 18
Figura 3.1 – Microcontrolador *ESP32 Development Kit* 19
Figura 3.2 – Modulo GY-LSM303C constituído pelo acelerómetro, magnetómetro e sensor de temperatura. 20
Figura 3.3 – Sistemas físicos. a) Motor trifásico de 0.37kW, b) Variador de frequência OMRON MX2. 20
Figura 3.4 – Transformadores de corrente. a) Transformador Efergy 90AS, b) Transformador ECS1030-L72. 21
Figura 3.5 – Circuito integrado LTC3588. 21
Figura 3.6 – Sistema de transporte, motor WAF10 DRN63MS4 de 0.12 kW 21
Figura 3.7 – Esquema do protótipo proposto. 22
Figura 3.8 – Locais de estudo. a) Primeiro local, b) Segundo local, c) Terceiro local – lateral. 23
Figura 3.9 – Peça 3D acoplada ao motor juntamente com os pesos/cargas. 23
Figura 3.10 – Imagem representativa da “Anomalia Obstáculo”, o obstáculo é visível a branco. 24
Figura 3.11 – Aumento da carga balanceada em modo “Normal” e modo “Anomalia Obstáculo”. a) Carga 1 – 1 b) Carga 2 – 2 c) Carga 3 – 3. 25
Figura 3.12 – Exemplo de uma linha de informação. A azul mostra os dados do acelerómetro e a laranja os dados do magnetómetro. 26
Figura 3.13 – Sequência de etapas efetuadas na ferramenta *Colab.* 27
Figura 3.14 – Diagrama do desenvolvimento da solução final. 29
Figura 3.15 – Exemplo real do fluxo de código em Node-RED criado para o projeto. 31
Figura 3.16 – Interface gráfica, a vermelho a informação referente do acelerómetro, a amarelo as classificações efetuadas e a verde os dados da temperatura. 32
Figura 3.17 – Etapas efetuadas no estudo do aproveitamento de energia. 32

Figura 3.18 – Locais de adquirição de energia. a) Posições ao redor do motor b) Acoplamento nos cabos de alimentação.....33
Figura 3.19 – Protótipo de *energy harvesting*. a) Entrada para o IC b) Armazenamento de energia c) Saída do IC. 34
Figura 4.1 – Leitura do sensor na posição a). 1 – Acelerómetro: Eixo X – Nº amostras, Y – mg. 2 – Magnetómetro: Eixo X – Nº amostras, Y – G (Gauss). 35
Figura 4.2 – Leitura do sensor na posição b). 1 – Acelerómetro: Eixo X – Nº amostras, Y – mg. 2 – Magnetómetro: Eixo X – Nº amostras, Y – G (Gauss). 36
Figura 4.3 – Leitura do sensor na posição c). 1 – Acelerómetro: Eixo X – Nº amostras, Y – mg, 2 – Magnetómetro: Eixo X – Nº amostras, Y – G (Gauss). 37
Figura 4.4 – Dados exemplos do acelerómetro nos diferentes modos de funcionamento. 38
Figura 4.5 – Evolução da *loss* e da *accuracy* ao longo do treino de 2000 *epochs*. 38
Figura 4.6 – Valor da *accuracy* com dados divididos para teste. 39
Figura 4.7 – Matriz de confusão sobre os dados de teste, com 2000 *epochs*. 40
Figura 4.8 – Evolução da *loss* e da *accuracy* ao longo do treino de 1000 *epochs*. 41
Figura 4.9 – Valor de *accuracy* utilizando os dados de teste. 41
Figura 4.10 – Matriz de confusão dos *dataset* de teste do modelo com 1000 *epochs*....42
Figura 4.11 – Código utilizado para a conversão do modelo em *tensorflow lite*. 43
Figura 4.12 – Matriz confusão da distribuição das classificações. 44
Figura 4.13 – Matriz de confusão final percentual. 45
Figura 4.14 – Aquisição dos transformadores de corrente. a) Transformador cinzento b) Transformador preto. 46
Figura 4.15 – Resultados dos transformadores de corrente posicionados em uma fase de alimentação. a) – Frequência de 10Hz; b) Frequência de 30 Hz; c) Frequência 50Hz. 47
Figura 4.16 – Tempo do primeiro carregamento. a) Transformador cinzento; b) Transformador preto. 48
Figura 4.17 – Tempo entre picos de saída. a) Transformador cinzento; b) Transformador preto. 49
Figura 4.18 – Resultado do transformador de corrente preto com osciloscópio na escala de 50 segundos. 50
Figura 4.19 – Região onde existe tensão disponível à saída. 51
Figura 4.20 – Gráfico do valor de tensão ao longo do tempo. 51

ÍNDICE DE TABELAS

Tabela 3.1 – Modo de funcionamento e distribuição da carga. 24
Tabela 3.2 – Parâmetros de treino da rede. 28
Tabela 4.1 – Resultado e distribuição das classificações. 43
Tabela 4.2 – Resultados obtidos através da análise efetuada. 52
Tabela 4.3 – Valores de energia ao fim de 1 hora e ao fim de 5 horas. 52
Tabela 4.4 – Resultados do funcionamento do sistema de minuto em minuto. 53
Tabela 4.5 – Resultados do funcionamento do sistema de 10 em 10 minutos. 54
Tabela 4.6 – Resultados do funcionamento do sistema de 30 em 30 minutos. 54
Tabela 4.7 – Resultados do funcionamento do sistema de hora em hora. 54
Tabela 4.8 – Resultados do funcionamento do sistema de 5 em 5 horas. 55
Tabela 4.9 – Resultado do sistema a funcionar de minuto em minuto. 55
Tabela 4.10 – Resultado do sistema a funcionar de 10 em 10 minutos. 56
Tabela 4.11 – Resultado do sistema a funcionar de 30 em 30 minutos. 56
Tabela 4.12 – Resultado do sistema a funcionar de hora em hora. 56
Tabela 4.13 – Resultado do sistema a funcionar de 5 em 5 horas. 56
Tabela 4.14 – Resultado sem uso *energy harvesting* e 5V de alimentação. 57
Tabela 4.15 – Resultado sem uso *energy harvesting* e 3.3V de alimentação. 58
Tabela 4.16 – Resultado com uso de *energy harvesting* e alimentação 5V. 59
Tabela 4.17 – Resultado com uso de energy harvesting e alimentação 3.3V. 59
Tabela 4.18 – Resultado do ciclo de minuto em minuto, *deep sleep* 10μA. 60
Tabela 4.19 – Resultado do ciclo a cada 10 minutos, *deep sleep* 10μA. 60
Tabela 4.20 – Resultado do ciclo a cada 30 minutos, *deep sleep* 10μA. 60
Tabela 4.21 – Resultado do ciclo de hora a hora, *deep sleep* 10μA. 61
Tabela 4.22 – Resultado do ciclo de 5 em 5 horas, *deep sleep* 10μA. 61
Tabela 4.23 – Resultado sem recurso a *energy harvesting*. 61
Tabela 4.24 – Resultado com recurso a *energy harvesting*. 62

1. INTRODUÇÃO

Os avanços a nível tecnológico são visíveis a cada dia que passa, no entanto, uma das áreas que mais sofreu evoluções foi a área de produção industrial com o objetivo de melhorar e facilitar a vida das pessoas. As mudanças ocorridas podem ser divididas em quatro grandes momentos históricos sendo elas as tão conhecidas revoluções industriais, Figura 1.1.

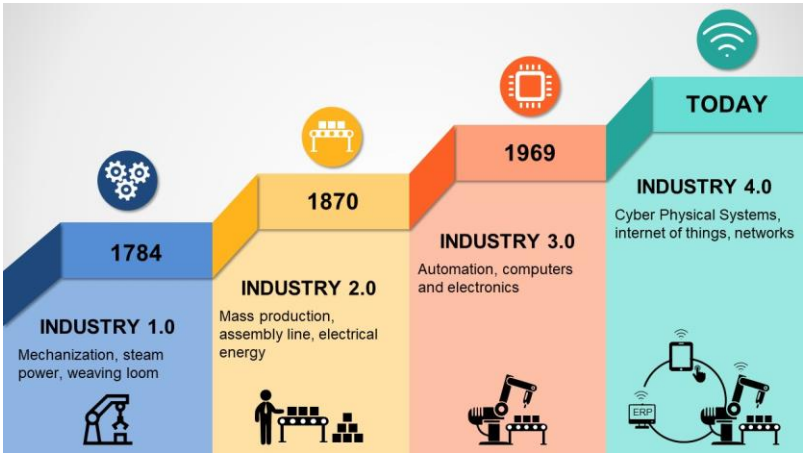


Figura 1.1 – Revoluções Industriais [1].

Abordando cada uma das revoluções industriais, a primeira revolução ocorreu durante o final do século XVIII e o início do século XIX e é conhecida por mecanizar os processos de produção, ou seja, é marcada pela invenção e implementação da máquina a vapor que provocou várias mudanças que permitiu aumentar a produtividade humana [2].

A segunda revolução deu-se passado um século comparativamente à primeira, nos finais do século XIX, carimbada pelo aparecimento de novas formas de energia (eletricidade, gás e petróleo) o que permitiu desenvolver a produção em massa, no entanto invenções como o automóvel e o avião são razões pela qual esta revolução ser considerada a mais importante[2].

Em meados do século XX testemunha-se a terceira revolução industrial, também conhecida como a revolução digital, dando início a exploração da energia nuclear e ficando marcada pela introdução da eletrónica, computadores e telecomunicações. Com a introdução das diferentes áreas mencionadas anteriormente e com a implementação de robôs e PLCs (*Programmable Logic Controllers*) permitiu elevar esta era para um alto nível de automatização [2].

Atualmente, vivemos a quarta revolução industrial sendo esta mudança conhecida por "Indústria 4.0". Esta revolução é considerada como uma otimização da terceira revolução, visto que os sistemas de produção são constituídos por tecnologia e informação sofrendo uma expansão com a comunicação de rede que permite a troca de informações entre todos os sistemas presentes. Áreas como a*rtificial intelligence* (AI), *Internet of Things* (IoT) e dados em tempo real estão associados a esta revolução [2].

1.1. Enquadramento

A manutenção de um equipamento, a nível industrial, é um aspeto importantíssimo e nos dias de hoje, é possível verificar a existência de diferentes tipos de manutenção, nomeadamente, manutenção proativa e manutenção reativa. Relativamente ao tipo reativo, a manutenção pode ser corretiva, tal como o nome indica acontece quando se corrige uma falha/problema no equipamento que causa a limitação na produção. Quando a avaria acontece a um nível intenso a manutenção é considerada de emergência.

No momento em que existe todo um plano por detrás da manutenção, a mesma é considerada proativa em que os métodos mais conhecidos são o preventivo e o preditivo. A manutenção preventiva é sem dúvida nos dias de hoje o tipo mais popular de manutenção proativa, em que se baseia em planear manutenções, que pode ter como base uso do equipamento, por exemplo, a quantidade de quilómetros ou horas de produção. No entanto apresenta algumas desvantagens pois existe o risco de serem agendadas manutenções em excesso visto que não se baseia nas condições reais dos equipamentos [3].

Na indústria, a manutenção preditiva (*predictive maintenance* – PdM) tornou-se bastante relevante na Indústria 4.0, visto que se baseia no acompanhamento periódico dos equipamentos através da análise de dados recolhidos de sensores. Nos dias de hoje este tipo de método tornou-se parte da "s*mart industrial maintenance*", antecipando possíveis falhas no equipamento através do uso de técnicas de "*machine learning*" [4]. Através da implementação da PdM os lucros tendem a aumentar e custos associados à manutenção sofrem uma redução em cerca de 25% e também uma redução de 40% no tempo em *downtime*, ou seja, tempo em que o equipamento de encontra inativo de forma não programada [5], [6].

Uma área complementar a ser abordada com este projeto é a área de *energy harvesting*, que consiste no processo de extração máxima possível de energia ambiente (energia desperdiçada), seguida de uma conversão para energia elétrica e do armazenamento da mesma para uso de carregamento do sistema. Existem várias fontes de extração de energia, sendo as mais conhecidas a extração através das vibrações (piezoelétrica), dos campos elétricos e eletromagnéticos, e também da energia solar.

Como foi referido anteriormente, existem diversas formas de extração, no entanto existem três componentes que são principais em todos os módulos [7]:

Harvester – A parte do sistema responsável pela conversão da energia ambiente em energia elétrica.

Interface Circuit – Responsável por aproveitar ao máximo a energia convertida tornando adequada para uso consoante a aplicação desejada.

Load – Associado a esta podem estar os dispositivos consumidores ou as componentes responsáveis por armazenar a energia.

A implementação deste processo fornece quantidade pequenas de energia e muitas vezes serve para prolongar a vida de equipamentos, de baixa energia, de pequenas dimensões, autónomos e *wireless*.

1.2. Identificação do problema

A nível industrial, no parque de máquinas existem facilmente um número elevado de equipamentos que necessitam de monitorização continua. No entanto é necessário processar uma enorme quantidade de amostras de diferentes sensores para analisar o funcionamento do equipamento e dessa forma esse processo fica dependente de sistemas de processamento de alta performance para a implementação de técnicas de *machine learning* (Figura 1.2 – a)), simultaneamente pode existir uma elevada transferência de dados na rede provocando um engarrafamento que pode resultar em perda de dados, que por sua vez pode levar ao aumento da complexidade do sistema global. Neste contexto um sistema de IoT com técnicas de avançadas de AI embebidas, permitirá processar localmente os dados em tempo-real e só enviar (se necessário) caraterísticas globais mais abstratas do estado do motor (Figura 1.2 – b)), o que pode ser uma solução para a redução de dados a serem transmitidos, mantendo a fiabilidade da monitorização.

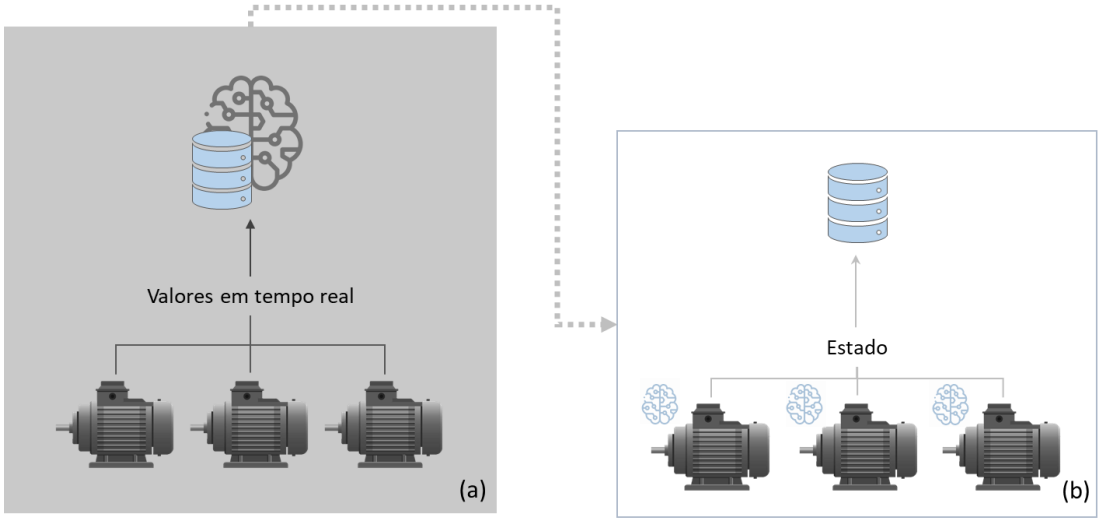


Figura 1.2 – Esquema de implementação. a) Transmissão em tempo real de todos os sistemas; b) Análise com *tinyML* local e transmissão do estado apenas quando necessário.

Neste contexto, muitas vezes estes sistemas implicam o uso de dezenas ou até mesmo centenas de baterias a necessitar de serem trocadas no mesmo espaço de tempo, o que em relação à manutenção torna os custos elevados (ex. tempo e material), além dos desperdícios se tornarem um subproduto poluente que muitas vezes não é reciclado da melhor forma.

A maioria dos sistemas embebidos implementados na indústria, no que diz respeito da sua alimentação, não são autónomos ou têm uma vida útil limitada, ou seja, estão dependentes de ligações de alimentação ou o uso de baterias com energia finita o que a nível industrial pode se tornar um problema para certos equipamentos, seja pelo uso de cabelagens ou a troca de bateria quando a mesma acaba.

Assim, de modo a reduzir e/ou eliminar estes tipos de problemas ou semelhantes, este projeto propõem a utilização de técnicas de *energy harvesting* complementado assim o sistema embebido como forma de lhe prolongar a bateria e o seu uso.

1.3. Objetivos

Pretende-se neste projeto criar um sistema embebido *wireless* capaz de adquirir dados em tempo real através de múltiplos sensores, com a capacidade de análise/classificação de anomalias de operação, aplicando uma rede *tinyML* (um tipo de *machine learning* para sistemas embebidos de pequenas dimensões e elevadas restrições) para obter uma classificação local do estado de funcionamento, e de forma que o sistema seja facilmente acoplado a um motor industrial existente em chão de fábrica.

De maneira a complementar este mesmo sistema, pretende-se ainda simular e estudar através de técnicas de extração de energia (*energy harvesting*) com o intuito de captar e utilizar energias próximas do sistema para carregar e/ou prolongar o *uptime* do mesmo, ou seja, estender a vida útil do equipamento. Desta forma, obtendo-se um sistema *plug-and-play* autónomo, com capacidade de análise e de classificação através técnicas avançadas de AI, testado e validado em laboratório. A Figura 1.3 representa em esquema o objetivo proposto.

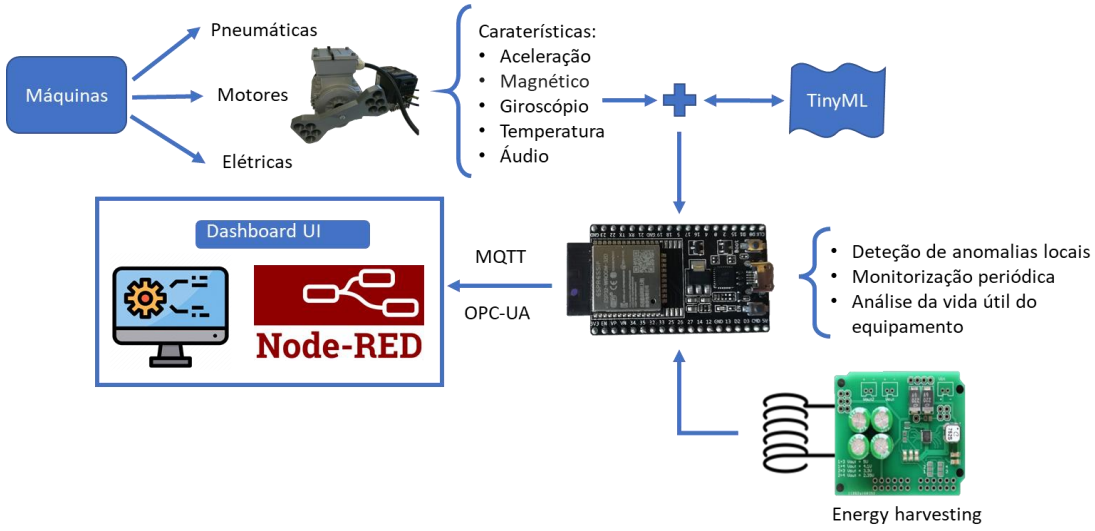


Figura 1.3 – Esquema o sistema proposto.

1.4. Disseminação do Trabalho

Durante o projeto foram realizados vários testes de divulgação pública, permitindo a divulgação do trabalho desenvolvido e a partilha do conhecimento adquirido com a comunidade científica.

- ***SAR – Symposium on Applied Research, 5th Edition*** – Apresentação e publicação de um *poster* sobre do trabalho planeado.
- ***SASYR – Symposium of Applied Science for Young Researchers, 1st Edition (Book of Abstracts)*** – Apresentação e publicação de um resumo do trabalho proposto no simpósio criado pelo centro laboratório ADiT (*Applied Digital Transformation Laboratory* do Instituto Politécnico de Viana do Politécnico), 2Ai (*Applied Artificial Intelligence Laboratory* do Instituto Politécnico do Cávado e do Ave) e CeDRI (Centro de Investigação em Digitalização e Robótica Inteligente do Instituto Politécnico de Bragança).

1.5. Estrutura do documento

O presente documento encontra-se estruturado em cinco capítulos, o primeiro capítulo conta com uma introdução ao projeto bem como a identificação do problema e quais os objetivos do mesmo. No segundo capítulo são descritos os conceitos teóricos bem como analisados trabalhos similares sobre tópicos principais referentes ao projeto.

De seguida, no terceiro capítulo é descrito as caraterísticas do *hardware* utilizado para a realização do projeto e de igual forma as etapas necessárias para a realização do mesmo. No quarto capítulo são enumerados os testes e simulações efetuadas e são também demonstrados os resultados obtidos ao longo do desenvolvimento do projeto.

Por fim, no quinto e último capítulo são apresentadas as conclusões em relação ao projeto e resultados obtidos, para terminar, são descritos possíveis trabalhos futuros de forma a complementar o sistema.

2. ESTADO DA ARTE

Neste capítulo é realizada uma análise de forma sucinta sobre diferentes métodos de aprendizagem e técnicas de *machine learning* bem como a utilização de *tiny machine learning* (ou *tinyML*) em sistemas embebidos. De seguida é abordada a evolução da utilização da manutenção preditiva para a deteção de anomalias em equipamentos industriais bem como a referência a trabalhos existentes. Por fim é abordado o progresso das técnicas de *energy harvesting*.

2.1. Técnicas de *machine learning*

A Humanidade, desde o início dos tempos, está em constante evolução e graças a sua criatividade tem se vindo a desenvolver diversas ferramentas e soluções de forma a tornar a sua vida mais fácil, nos dias de hoje umas das ferramentas mais faladas é o *machine learning* (ML). O ML é um subcampo da *artificial intelligence* (AI), que se foca no estudo de algoritmos computacionais que permitem lidar com dados de forma mais eficiente, uma vez que em diversas situações não temos a capacidade de interpretar a informação presente nos mesmos. Na indústria, existe cada vez mais dados à disposição o que têm levado a um aumento do uso de técnicas de ML pelas próprias empresas [8].

Existem diferentes tipos de técnicas de ML sendo que o *supervised learning* e o *unsupervised learning* são consideradas as técnicas mais importantes, existindo ainda a técnica de *reinforcement learning* (RL) [8], Figura 2.1.

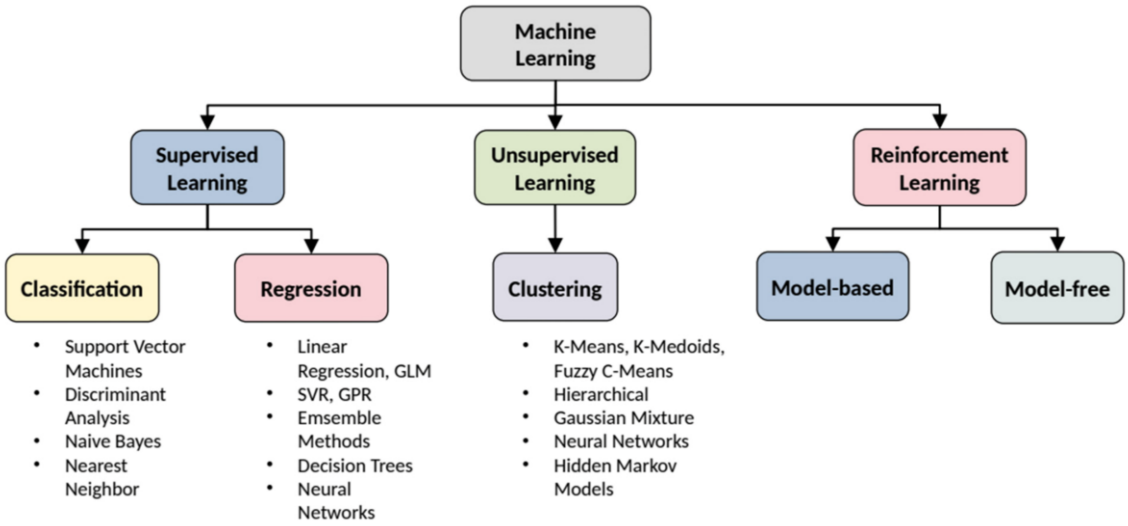


Figura 2.1 – Exemplo de algoritmos ML [9].

2.1.1. Técnicas de *supervised learning*

O *supervised learning*, a aprendizagem supervisionada, é a técnica mais comum utilizada, neste ambiente os dados recolhidos estão anotados, ou seja, aos dados/*inputs* é lhes associando um rótulo/*label*. Em outras palavras, os dados são associados a um atributo, por exemplo o bom funcionamento do motor e o mau funcionamento. Neste processo um *dataset* (conjunto de dados já anotados em vários estados) de entrada é então dividido em treino e em teste, o *dataset* de treino tem então associado a si um valor de output que é utilizado para treinar e ajustar o modelo pretendido, uma vez obtido o modelo o *dataset* de teste é utilizado para validar o mesmo, através da *accuracy*. O principal objetivo do modelo supervisionado é prever outputs, ou seja, as *labels* para cada novo input [8], [10], [11].

Os algoritmos mais comuns no *supervised learning* são as *Decision Trees*, o método de *Naïve Bayes* e *Support Vector Machines*. *Decision Tree* é uma abordagem de *supervised machine learning e* é das abordagens mais antigas e proeminentes.

Uma *decision tree* modela a lógica de decisão, ou seja, determina o resultado de forma semelhante à estrutura de uma árvore, geralmente começa com um único "nó", que se divide num possível resultado na forma de sim ou não que dita qual dos caminhos (os "ramos") a seguir até ao próximo "nó", Figura 2.2. A *decision tree* tem como vantagens, por exemplo, a facilidade de interpretação, rapidez de serem aprendidas, a adaptabilidade de se adicionar novas opções. Por outro lado, árvore de decisão pode ser tornar instável, pode ser difícil controlar o tamanho da árvore o que pode levar a uma elevada complexidade[8], [10]–[12].

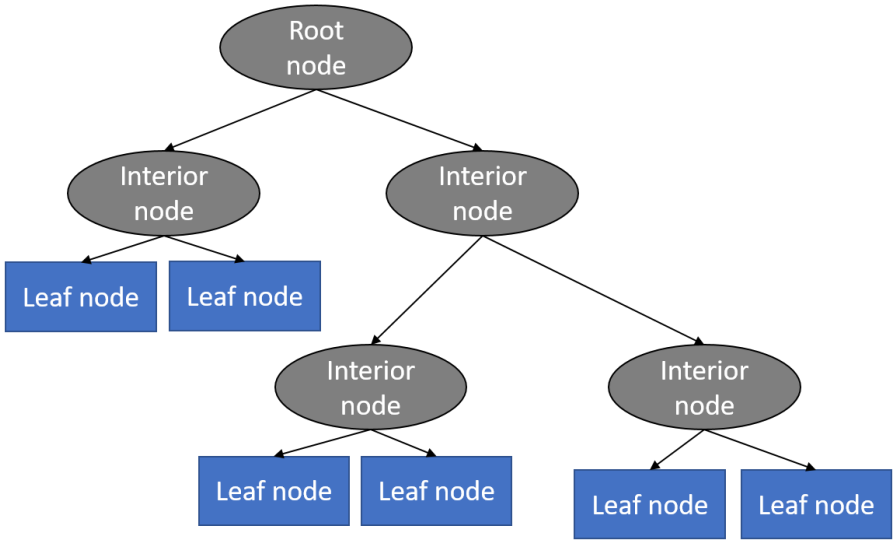


Figura 2.2 – Estrutura da *decision tree* [13].

Outro algoritmo que pertence ao grupo do *supervised learning* é o algoritmo *Support Vector Machine* (SVM). O objetivo de uma SVM é encontrar um hiperplano num espaço "N" – dimensional (onde "n" é o número de caraterística), no entanto existe uma vasta possibilidade de hiperplanos onde o propósito é separar as diferentes classes maximizando a margem entre as mesmas, diminuindo assim o erro associado à

classificação. Por margem máxima entende-se o afastamento máximo entre os pontos de dados de ambas as classes, os pontos mais próximos do hiperplano são considerados *support vectors*, Figura 2.3, são esses pontos que influenciam as caraterísticas do hiperplano [8], [10]–[12].

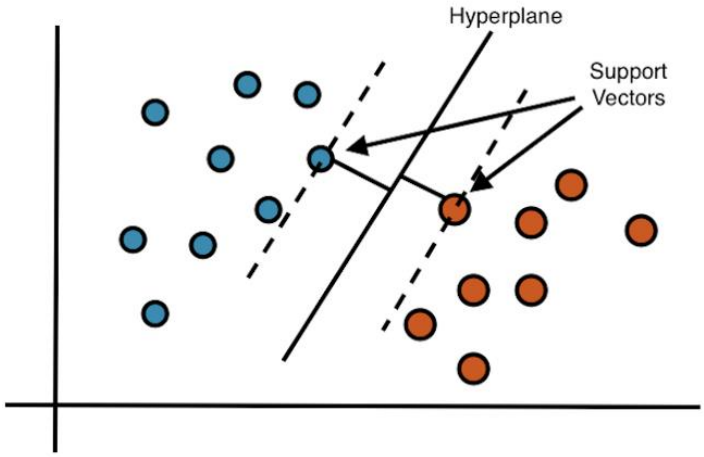


Figura 2.3 – Exemplo de hiperplano em *Support Vector Machine* [14].

Por fim, o algoritmo *Naïve Bayes* (NB) é uma técnica de classificação que se baseia, tal como o nome indica, no teorema de *Bayes* que se baseia no princípio de funcionamento da probabilidade condicionada, Equação 1.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \qquad (Eq.\,1)$$

O teorema de *Bayes* afirma que é possível determinar a probabilidade de "A" acontecer sabendo que "B" ocorreu. Este modelo tem como nome "naïve" devido a supor a independência entre todos os atributos dos dados que no mundo real isso sabe-se que dificilmente acontece [8], [12].

2.1.2. Técnicas de *unsupervised learning*

Outro método de ML é o *unsupervised learning*, a aprendizagem não supervisionada, que ao contrário do *supervised learning* não necessita de supervisão, ou seja, aos dados de entrada não lhes é correspondido nenhum tipo de *label* para mais tarde o modelo prever. O principal objetivo deste método é descobrir semelhanças e relações entre os dados agrupando-os dessa mesma forma[8].

As técnicas mais populares deste método são a análise dos componentes principais, o *principal component analysis* e o *k-means clustering*.

O algoritmo *principal component analysis*, também conhecido como PCA, é um algoritmo estatístico utilizado para converter variáveis correlacionadas em um número menor não correlacionadas chamadas de componentes principais. O objetivo é reduzir as dimensões de um vasto número dados de *input*, que por sua vez torna difícil a identificação, sem perder informações críticas sobre os mesmos recriando um novo conjunto

de dados. Contribui ainda para a compactação dos dados, extração de caraterísticas sobre os dados e também para a visualização dos mesmos [8], [15], [16].

Uma abordagem simples, mas amplamente utilizada no *unsupervised learning* é o *k-means clustering*, Figura 2.4, no entanto antes é preciso compreender em que consiste o *clustering*. O *clustering* traduz-se no agrupamento de diversos dados de caraterísticas semelhantes sem obter conhecimento prévio de *labels* sobre o conjunto, aos grupos criados dá-se o nome de *clusters* [17].

Abordando o algoritmo de *k-means clustering*, em que "k" significa o número de centroides/clusters que se pretende obter do *dataset*, no entanto é necessário ter em atenção o número escolhido pois a quantidade de dados presentes dentro dos *clusters* pode aumentar ou diminuir consoante o número pretendido dos mesmos. A criação do *cluster* passa por diferentes partes, sendo a primeira etapa a escolha aleatória dos pontos centrais dos "k" *clusters*¸ de seguida é feita a atribuição dos dados ao respetivo *cluster*, ou seja, é associado o dado ao grupo que tiver menor distância referente ao centroide. Por fim é calculado um novo centroide para o *cluster* através da média da posição de todos os valores dentro do mesmo, repete-se os últimos passos até se atingir os pontos centrais ideais [8], [15], [17].

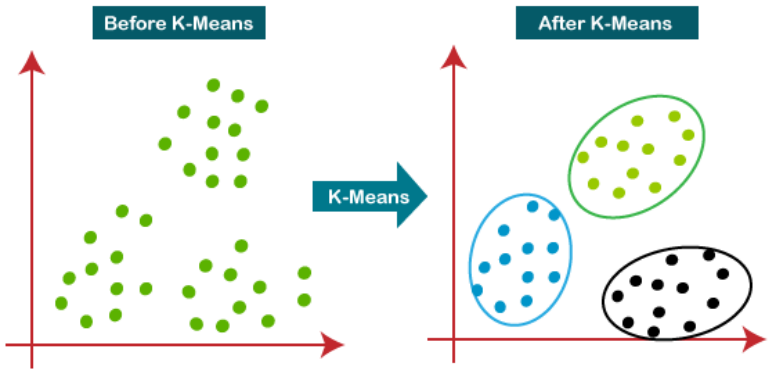


Figura 2.4 – Exemplo de algoritmo *K-Means Clustering* [18].

2.1.3. Técnica de *reinforcement learning*

Um método presente na área da ML é também o *reinforcement learning* (RL), Figura 2.5, que se concentra na autoaprendizagem do sistema, ou seja, através de um conjunto de tentativas e erros, determina uma solução através de um mecanismo de recompensas ou punições sempre tendo como objetivo maximizar a recompensa total e minimizar o risco/erro. O algoritmo de RL, conhecido como agente, está em constante aprendizagem com o ambiente envolvente de forma interativa, descobrindo quais as ações lhe recompensam mais [8], [19].

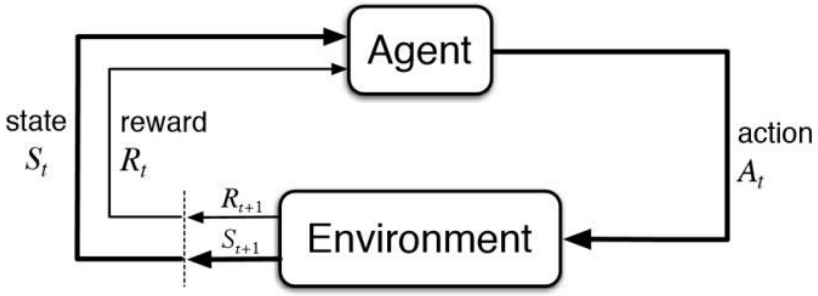


Figura 2.5 – Diagrama básico de *reinforcement learning* [20]**.**

2.1.4. Técnicas de *neural networks*

O *deep learning* (DL) é um subtema do ML que tem como base o uso de *artificial neural network* (ANN). As ANNs, geralmente são chamadas e tratadas por apenas de *neural network* (NN) por causa de imitarem o funcionamento do cérebro humano, são constituídas por uma série de algoritmos que procuram identificar relações entre o conjunto de dados, as NN podem ainda adaptar-se a mudanças de *input* com o intuito de obter o melhor resultado sem ser necessário modificar os critérios do *output*. Estes algoritmos são constituídos pelo menos por três tipos de camadas, conhecidas como *layers*, sendo a primeira a camada de entrada ou *input layer* onde recebe os dados de entrada, as camadas intermédias chamadas de camadas ocultas ou *hidden layers* é responsável por processar o *input* atribuindo diferentes pesos a cada neurónio, esta camada pode atingir diversas proporções, ou seja, quanto maior o número de *hidden layers* mais complexa e "*deep*" é a rede. A camada final corresponde a camada de saída ou *output layer* que fica responsável por emitir o *output* ao problema [8], [21]. A Figura 2.6 serve como exemplo de estrutura de uma *neural network*.

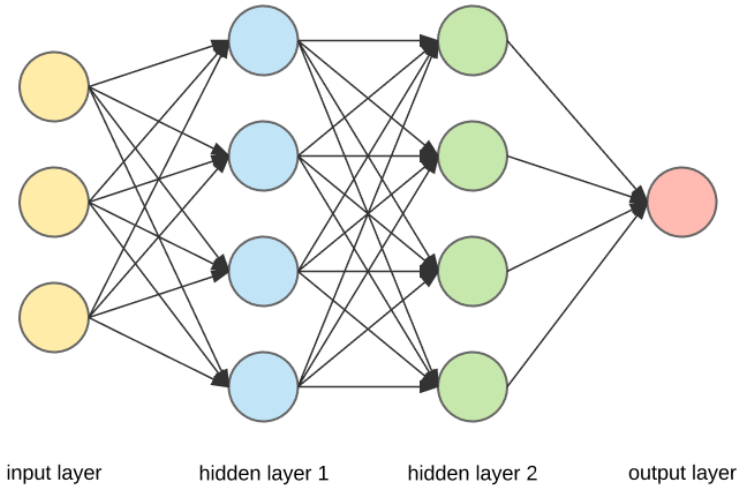


Figura 2.6 – Exemplo de NN [22]**.**

Existem diferentes tipos de *neural networks*, *feed forward neural networks* (FFNN) *onde* cada "nó" está conectado a um nó na *layer* seguinte estando todos os nós conectados totalmente, *recurrent neural networks* (RNNs) que são uma variação das redes FF, no entanto tem a caraterística de receber um *input* atrasado num tempo específico com o intuito de aceder a informação passada, e as redes neuronais convolucionais (*convolution neural networks* – CNNs) [23], [24].

As redes convolucionais têm por base as redes FF, no entanto utiliza princípios de álgebra linear, mais especificamente, o uso de matrizes contando ainda com o uso das camadas de convolação e de *pool*. Por convolução entende-se a passagem dos dados por um filtro, também conhecido por *kernel*, multiplicando o mesmo de forma a reduzir a quantidade dos dados sem perder as caraterísticas [23], [25]. A Figura 2.7 serve como exemplo para uma melhor compreensão do processo de convolução.

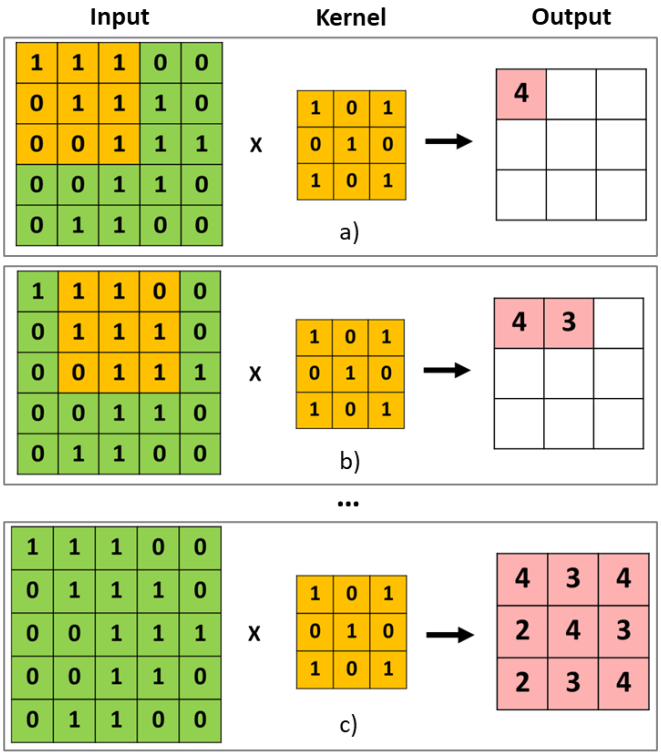


Figura 2.7 – Exemplo de Convolução. a) 1º Passo b) 2º Passo c) Resultado final [26].

Após a camada de convolução existem outras operações disponíveis em ML. Uma das mais comuns é uma camada de *pool* com o objetivo de reduzir o número dos dados, esta operação não é feita através de uma operação matemática, mas sim por regras dependendo do tipo de *pool* utilizado.

Existem dois tipos de *pool*, Figura 2.8, o *max pooling* retorna o máximo valor dos valores que o *kernel* está a estudar, por fim o outro tipo de *pooling* é o *average pooling* no qual o valor retornado corresponde ao valor médio dos valores presentes dentro do *kernel* [23], [25].

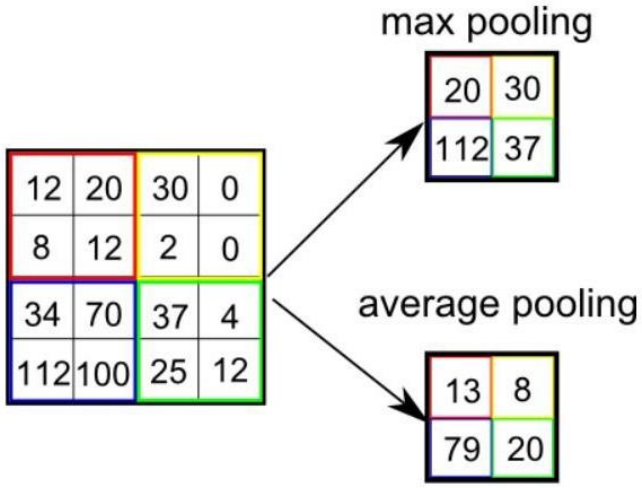


Figura 2.8 – Tipos de *pooling* [26]**.**

É importante referir que as técnicas de ML não são apenas as referidas anteriormente, mas que existem ainda diferentes técnicas no mundo do *machine learning*, no entanto, não foram mencionadas devido a não serem tão relevantes para o desenvolvimento do projeto.

2.2. *Tiny machine learning*

O *machine learning* tem apresentado um aumento no seu uso e na sua otimização ao longo dos tempos, em contrapartida o custo computacional associado ao treino dos modelos em crescido e por essa razão a maioria das técnicas de ML são implementadas em servidores e supercomputadores localizando-se na *"cloud"*.

2.2.1. Introdução ao *tinyML*

Um tema que tem sofrido um grande desenvolvimento nos últimos anos é o uso de sistemas embebidos baseados em microcontroladores, devido ao seu baixo consumo, desempenho e fiabilidade associada [27]. Um microcontrolador é um circuito integrado, *integrated circuit* – IC, constituído por um núcleo de processador, por memórias voláteis e não voláteis e disponibiliza também periféricos de entrada e saídas.

Os microcontroladores têm sido adotados como uma peça central servindo de *edge device* na interligação na área de IoT para a recolha de dados sensoriais e envio dos mesmo para a *"cloud"*. Facilmente se consegue atingir uma enorme quantidade de dados referente a todos os dispositivos IoT instalados, o que de certa forma irá agravar e dificultar a transferência de todos os dados para a *"cloud"* para o processamento do ML. De modo a minimizar os riscos associados à transferência de dados uma solução possível é integrar o chamado *tinyML* [27].

TinyML é uma vertente em crescimento dentro do *machine learning* que interliga as técnicas de aprendizagem com sistemas embebidos, Figura 2.9, dessa forma eliminando a necessidade de aceder frequentemente ao serviço que se encontra na *"cloud"*. A implementação localmente destas técnicas permite

que os dispositivos de IoT possam não só recolher dados de sensores, mas também processar e analisar com um algoritmo ML embutido no equipamento [27]–[29].

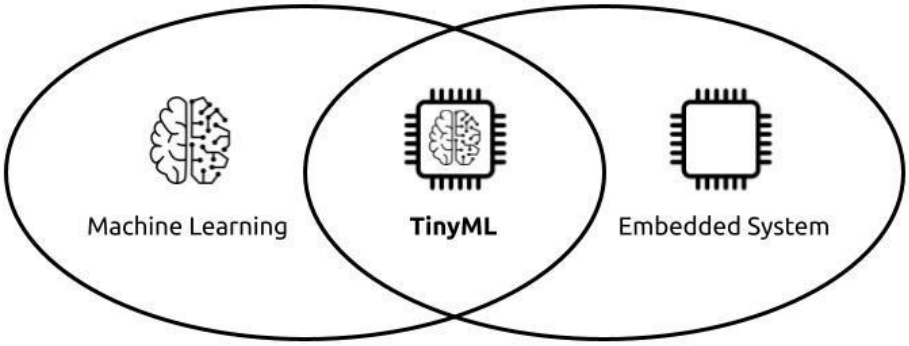


Figura 2.9 – *Machine learning* em sistemas embebidos [30]**.**

2.2.2. Benefícios do *tinyML*

O *tinyML* é descrito por diferentes características que de uma forma ou de outra servem de comprovativo para a sua eficácia e o seu uso ser indispensável. As caraterísticas apresentadas a seguir servem com um exemplo [27], [28]:

Network bandwidth (Largura de banda de rede) – Os dispositivos IoT que tem por base o uso de *tinyML* oferece independência permitindo uma baixa transmissão e uma largura de banda menor, visto que os dados não precisam de ser enviados constantemente, uma vez que são processados e analisados previamente.

Security and privacy (Segurança e Privacidade) – Um aspeto que torna difícil a aprovação do uso de dispositivos IoT é a segurança e a privacidade dos dados, pois são grandes quantidades de dados "privados" a ser enviados o que abre a possibilidade a ataques cibernéticos. Com o *tinyML* os dados são enviados apenas quando necessário, estando menos expostos aumentando a segurança e a privacidade dos mesmos.

Latency (Latência) – Uma das etapas de um dispositivo IoT é a transferência dos dados recolhidos dos sensores para a "*cloud"*, essa transferência termina muitas vezes com a entrega do *output* da decisão no dispositivo. Todo este processo tem um tempo de duração (latência) associado, o que em sistemas críticos como veículos autónomos pode ter consequências enormes, através do *tinyML* a comunicação externa deixa de ser necessária obtendo uma latência reduzia perto de zero.

Energy efficiency (Eficiência energética) – A eficiência energética é uma caraterística importante em microcontroladores, uma vez que estes dispositivos utilizam tipicamente baterias/pilhas para estarem operacionais. Deseja-se que os sistemas IoT funcionem o maior tempo possível o que implica que em certas alturas os mesmos se encontrem num estado de *deep sleep*, no entanto o serviço de transmissão dos dados pode

consumir mais energia do que a própria recolha dos dados. Ao remover da equação os envios desnecessários o *uptime* do módulo de IoT aumenta.

Low cost (Baixo custo) – O uso de sistemas à base de microcontroladores são simples e de *hardware* acessível, que combinado com a flexibilidade de programação torna estes sistemas numa opção ideal para o tinyML, com uma ampla gama de aplicações.

Atualmente, como já foi referido anteriormente, o uso de dispositivos IoT de baixo consumo que juntamente com as vantagens do *tinyML* estão cada vez mais presentes no dia a dia e em diversas áreas. Na área da saúde com a monitorização da frequência cardíaca, oxigenação do sangue, etc., a partir da leitura de diversos sensores e fornecendo uma representação dos mesmos dados em tempo real [28], [31].

Um outro exemplo onde o *tinyML* é integrado, é na indústria para a monitorização e o controlo de forma a assegurar a qualidade das linhas de produção e também de forma a elaborar diagnósticos sobre as máquinas em tempo real, o que irá levar à redução de custos [28], [31].

2.3. Deteção de anomalias em equipamentos indústrias

Com o intuito de atingir um desenvolvimento mais completo foi efetuada uma revisão bibliográfica sobre de um dos assuntos com maior peso no projeto, a predição de falha em motores elétricos.

Os motores de indução alternados são a grande escolha relativamente ao uso na indústria, no entanto não estão imunes a sofrer falhas e avarias, falhas essas que podem ser classificadas em diferentes categorias, mecânicas, associadas a defeitos nos rolamentos, excentricidade e falhas na carga do motor. Existe também falhas do tipo elétricas, mais relacionadas a problemas de alimentação e do rotor e estator do motor [32], [33].

Existem vários métodos para acompanhar/monitorizar os motores em questão, sendo eles através da vibração da carcaça do estator, através do som porque peças de ferro expostas a um campo magnético estão sob efeito de stress de Maxwell, que produz ruído eletromagnético (EM) que sobre o efeito de falhas esse ruído tende a aumentar em 12dB. A corrente é uma caraterística também utilizada no diagnóstico de falhas tendo em conta, que num motor em funcionamento ideal a corrente consumida é constante, falhas ao nível do estator produzem oscilações significativas no fluxo magnético tornando-a uma excelente caraterística de análise [32], [33].

Porém é necessário proceder a um tratamento dos dados, esse pode ser ao nível do cálculo do valor RMS da tensão, no domínio do tempo, no domínio da frequência através do uso de *Fast Fourier Transform* (FFT), ou ao nível de tempo-frequência com técnicas da espectrograma, *Short Time Fourier Transform* (STFT) ou *Wavelet Transform* (WT) [32]–[34].

No artigo [35], os autores estudaram o funcionamento dos motores em termos da corrente e do fluxo magnético envolvido nos mesmos. Numa abordagem inicial utilizaram o método dos elementos finitos para o estudo de três motores de caraterísticas bastante semelhantes e do impacto provocado pelo tamanho e o número de ranhuras do rotor no que diz respeito à frequência de velocidade do motor. Em relação aos testes, dos motores simulados, num dos testes o motor encontrava-se saudável e os restantes possuíam barras do rotor

estragadas, para a extração e análise das caraterísticas foi utilizada a FFT, no qual se verificou um aumento das amplitudes das frequências à medida que o número de falhas no rotor aumenta. Os mesmos testes foram efetuados em ambiente experimental obtendo resultados semelhantes.

Em [36], os autores propuseram um sistema de monitorização para deteção de anomalias utilizando CNN 1D compactas, sendo estas mais utilizadas para reconhecimento de imagens ou vídeos, o método foi desenvolvido para a extração e classificação de valores da corrente do motor sem sofrerem pré-processamento. A nível experimental utilizaram 260 motores tanto "saudáveis" como com defeito, as métricas de eleição foram a *accuracy*, *sensitivity*, *specificity* e *positive prediction*, dividindo ainda em duas classes distintas *"Healthy"* e *"Faulty"* e no qual os resultados obtidos demostram uma precisão de 97%, demonstrando a sua viabilidade.

2.4. *Energy harvesting*

Um tópico também com elevada importância para o desenvolvimento do projeto juntamento com o anterior é o *energy harvesting*. Para uma maior compreensão sobre o assunto foi efetuado uma revisão bibliográfica sobre as técnicas de *energy harvesting* e o uso das mesmas.

Em 1826, Thomas Johann Seebeck regista a primeira forma de gerar energia através de *energy harvesting* com duas placas de metais diferentes a temperaturas diferentes, o efeito termoelétrico. Anos mais tarde Pierre e Jacques Curie provaram a técnica piezoelétrica, ou seja, que certos cristais demostravam energia quando eram sujeitos a algum tipo de *stress* [37].

Atualmente com a implementação de sensores em sistemas *wireless* na área de IoT e com todo o progresso no que diz respeito aos dispositivos de monitorização, os mesmo continuam a ter uma pequena limitação que é o uso de baterias como fonte de energia visto que a energia disponível presente na bateria ainda é limitada o que torna vida útil do equipamento finita. A substituição de baterias é uma solução viável no que diz respeito a sistemas IoT de pequenas dimensões, contudo em sistemas de grandes dimensões tem um custo de manutenção elevado o que torna essa solução ineficiente. Uma área que tem estado em desenvolvimento e que pretende ser uma solução promissora é a utilização e aplicação de técnicas de *energy harvesting* [38] nos sistemas de IoT.

O *energy harvesting* é um processo de aproveitamento da energia do meio envolvente com o intuito de alimentar o dispositivo IoT e aumentar o seu *uptime*. O sistema de *energy harvesting* pode comportar-se de duas formas [38], [39]:

Harvester–Use – A energia aproveitada é diretamente usada para alimentar o sistema, caso a energia não for o suficiente o mesmo era continuar no estado limitado.

Harvester–Store–Use – A energia num primeiro passo é guardada para ser usada futuramente se for necessário.

2.4.1. Fontes de *energy harvesting*

No que diz respeito ao *energy harvesting* existem diferentes formas e fontes de energia no qual se pode aplicar as suas técnicas. As fontes podem ser consideradas de fontes naturais, energias ambientais existentes no meio envolvente, no entanto, existem também fontes artificiais que são geradas pelo ser humano e/ou pelas suas atividades [39].

Fontes de energia solar e eólica são altamentes eficientes, para a captação de energia solar recorre-se ao uso de células fotovoltaicas que convertem os raios luminosos em energia elétrica, no caso da eólica o vento é utilizado como forma de adquirir energia elétrica [38]–[40]. O uso do calor e a variação de temperatura como forma de gerar energia é também muito conhecida, na energia termoelétrica existem duas técnicas associadas, o efeito termoelétrico e o piroelétrico. A técnica termoelétrica converte diretamente as diferenças de temperatura em energia elétrica, ou seja, entre uma superfície quente e outra fria, enquanto a técnica piroelétrica gera energia consoante as mudanças de temperatura, ou seja, utiliza um cristal piroelétrico e quando o mesmo é aquecido ou arrefecido gera energia elétrica [38]–[40].

Uma outra fonte de *energy harvesting* é a energia mecânica que pode ser dividi em energia cinética, que está relacionada com o movimento de um objeto, ou em energia potencial relacionado com a posição de um objeto ou ao trabalho realizado pelo mesmo, as técnicas utilizadas para captar este tipo de energia são as piezoelétricas, electroestáticas e eletromagnéticas. Na técnica piezoelétrica é utilizado um material específico capaz de converter energia cinética diretamente em energia utilizável quando o mesmo é sujeito qualquer tipo de stress, a electroestática é baseada na mudança de capacitância e por fim a técnica eletromagnética utiliza, tal como o nome indica, o princípio de indução eletromagnética, ou seja, o uso do campo magnético [38]–[40].

O *energy harvesting* é também um tópico importante devido as vantagens que oferece na sua capacidade autossustentável e de redução da pegada de carbono, neste sentido foi efetuada uma pequena revisão bibliográfica de estudos sobre a implementação das técnicas de *energy harvesting*.

No artigo [41], avaliaram a implementação de uma tecnologia de capaz de captar energia magnética, chamada de *magneto-mechano-electric* (MME) que interliga o uso de materiais piezoelétricos (ex. para uso em vibração) juntamente com o aproveitamento de campos magnéticos existentes no meio ambiente. Abordam também a ideia de desenvolver geradores híbridos, ou seja, geradores MME juntamente com diferentes tipos de geradores, podendo ser eles magnéticos, solares ou termoelétricos com o intuito de gerar uma maior quantidade de energia para disponibilizar, por exemplo, a sistemas de sensores *wireless.*

Em [42], é feito um estudo relativamente ao uso dos campos elétricos e magnéticos como forma de aproveitamento de energia da rede elétrica. Focando-se na parte do campo magnético o estudo foi dividido em dois tipos, o *low potential* e o *high potential*, essa divisão é associada com o potencial do próprio sistema têm. A diferença entre o *low* e o *high* é maioritariamente a estrutura e o local de montagem, ou seja, a solução de *low potential* é implementada com alguma distância do cabo elétrico enquanto a solução de *high potential* é implementada em torno do mesmo. O esquema de montagem é possível ser analisado na Figura 2.10.

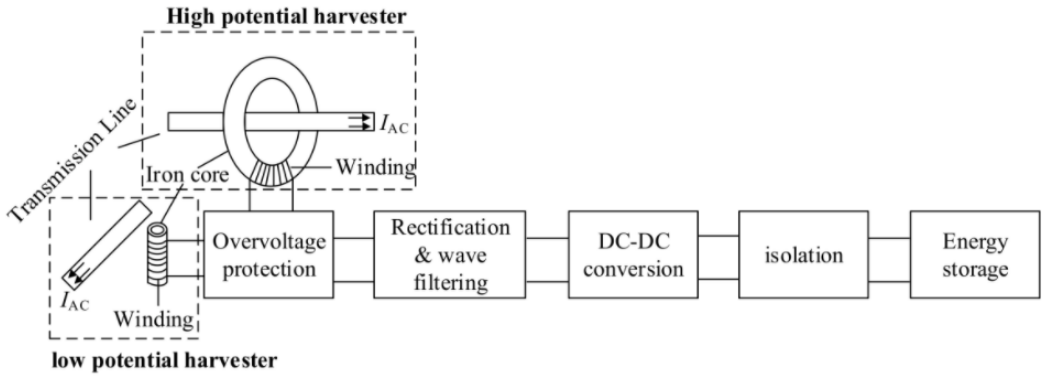


Figura 2.10 – Esquema de captação de energia magnética [42]**.**

No artigo [43], os autores fazem um estudo de desenvolvimento de dois *step-up converters* de baixa tensão utilizando dois circuitos integrados LTC3108 e o LTC3109. O LTC3108 necessita de uma tensão de 20mV onde apenas trabalha com *input* positivo enquanto o LTC3109 trabalha com *inputs* positivos e negativos. Efetuou-se a medição da relação da tensão de entrada (V*in*) com a corrente de saída (I*out*) limitando a tensão de saída a 4V. Nas medições do conversor com base no LTC3109 utilizaram energia termoelétrica proveniente de um dedo humano em contacto com uma bateria térmica, verificando que o conversor permitia o funcionamento com diferenças de temperatura de 0.1 K (kelvins) capaz de acender um LED (*Light-Emitting Diode*).

3. DESENVOLVIMENTO

Neste capítulo são abordadas as etapas desenvolvidas ao longo da elaboração do projeto. Inicialmente será feita uma abordagem teórica das caraterísticas sobre o hardware utilizado no projeto, de seguida é apresentado o desenvolvimento referente ao algoritmo de classificação descrevendo-se as fases de criação do *dataset* e a elaboração da rede de classificação. Este capítulo foca também a implementação do sistema de "manutenção", da monitorização dos dados e como ela é efetuada e quais as ferramentas utilizadas. Para terminar é descrito ainda o desenvolvimento em relação à parte do *energy harvesting*.

3.1. Ferramentas de *hardware*

A primeira etapa de desenvolvimento foca-se no *hardware* utilizado, como já foi referido anteriormente, o projeto divide-se em duas áreas, em relação ao *hardware* utilizado a mesma divisão pode ser aplicada.

Em relação à secção correspondente a aquisição de dados bem como a todo o processamento necessário, foi utilizado como microcontrolador a placa *ESP32 Development Kit* (*ESP32 DevKitC* – Figura 3.1), devido à mesma permitir o uso de WiFi, disponibiliza ainda 4 Megabytes de memória *flash*, entre outras caraterísticas. Um atributo que se realça é a existência de diferentes configurações em relação ao modo de energia um aspeto importantíssimo na interligação dos dois grandes temas do projeto. Existe ainda uma vasta comunidade a suportar este equipamento.

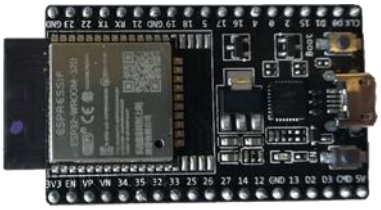


Figura 3.1 – Microcontrolador *ESP32 Development Kit*.

Sobre o sensor utilizado para a captação dos dados de interesse utilizou-se o modulo integrado GY-LSM303C, Figura 3.2, que apresenta seis graus de liberdade (DoF), devido a integrar um acelerómetro de três eixos juntamente com um magnetómetro, LIS3MDL, igualmente de três eixos. O integrado apresenta uma grande flexibilidade, em relação ao acelerómetro, o mesmo permite uma escala de sensibilidade variável de ±2g, ±4g, ±6g ou ±8g, enquanto o magnetómetro usa uma escala de ±16 gauss, como protocolo de comunicação o módulo suporta o protocolo de I2C (*Inter-Integrated Circuit*) e o protocolo SPI (*Serial Peripheral Interface*). Adicionalmente conta ainda com um sensor de temperatura.

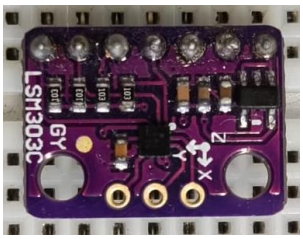


Figura 3.2 – Modulo GY-LSM303C constituído pelo acelerómetro, magnetómetro e sensor de temperatura.

Por fim, todos os elementos anteriores não fazem sentido sem o equipamento de teste onde acoplar o protótipo, neste caso utilizou-se um motor AC trifásico, Figura 3.3 – a), que à frequência da rede elétrica de Portugal (50Hz) disponibiliza uma potência de 0.37kW, com um fator de potência de 0.75, por fator de potência entende-se a razão entre a potência ativa do motor e a potencia, ou seja, entre a potência consumida para realizar movimento, o calor e etc., e a potência total entregue ao motor. Este tipo de motores possibilita duas formas de ligação sendo elas em estrela ou em triangulo, neste caso a ligação que foi utilizada foi ligação em estrela apresentando uma corrente de 1.06A e uma tensão de 380 – 415V, em carga.

De forma a colocar o motor em funcionamento foi utilizado o variador de frequência OMRON 3G3MX2-A2004-E, Figura 3.3 – b), facilitando a variação das frequências de trabalho do motor, assim como o seu ligar e desligar.

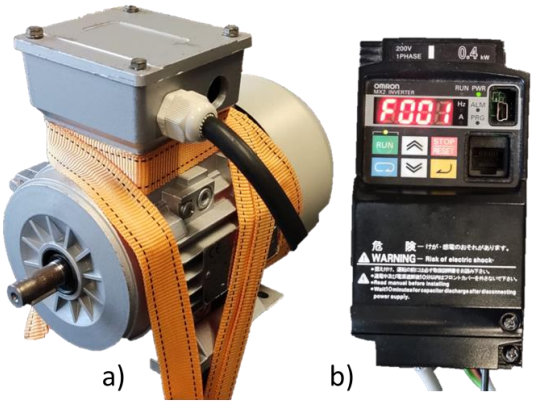


Figura 3.3 – Sistemas físicos. a) Motor trifásico de 0.37kW, b) Variador de frequência OMRON MX2.

O desenvolvimento conta ainda com uma segunda parte do projeto que é referente ao estudo do *energy harvesting*, para a realização desse estudo utilizou-se para o aproveitamento de energia dois transformadores de corrente. O primeiro transformador Efergy 90AS, Figura 3.4 – a) conta com uma razão de transformação de 1:1350 enquanto o segundo transformador ECS1030-L72, Figura 3.4 – b) conta com uma razão de 1:2000.

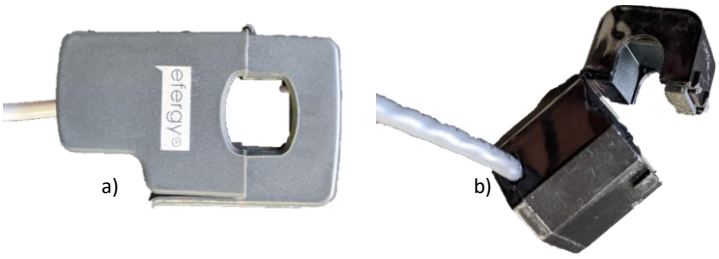


Figura 3.4 – Transformadores de corrente. a) Transformador Efergy 90AS, b) Transformador ECS1030-L72.

De forma a captar e gerir a energia aproveitada e gerada pelas pinças escolheu-se o circuito integrado LTC3588, Figura 3.5, recomendado para aplicações piezoelétricas bem como eletromecânicas visto que conta com um retificador de onda completa. O circuito permite acumular uma certa quantidade de corrente/tensão no condensador até atingir o valor que possibilita a sua transferência para a saída. O LTC3588 funciona entre os 2.7V e os 20V como tensão de entrada e disponibiliza um *output* estável selecionável entre 1.8V, 2.5V, 3.3V e de 3.6V.

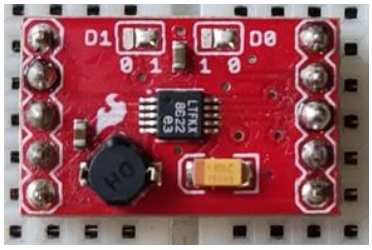


Figura 3.5 – Circuito integrado LTC3588.

Uma vez que o motor utilizado para a primeira parte do projeto encontrava-se em vazio, ou seja, não era possível aumentar o consumo de corrente. De forma a conseguir um maior consumo utilizou-se um *conveyor system* (sistema de transporte), Figura 3.6, constituído por um motor trifásico e uma caixa redutora, um sistema muito frequente em ambiente industrial permitindo assim simular de forma mais realística as condições onde o sistema possa ser inserido. O motor acoplado ao sistema de transporte é um WAF10 DRN63MS4 que conta 0.12Kw de potência.

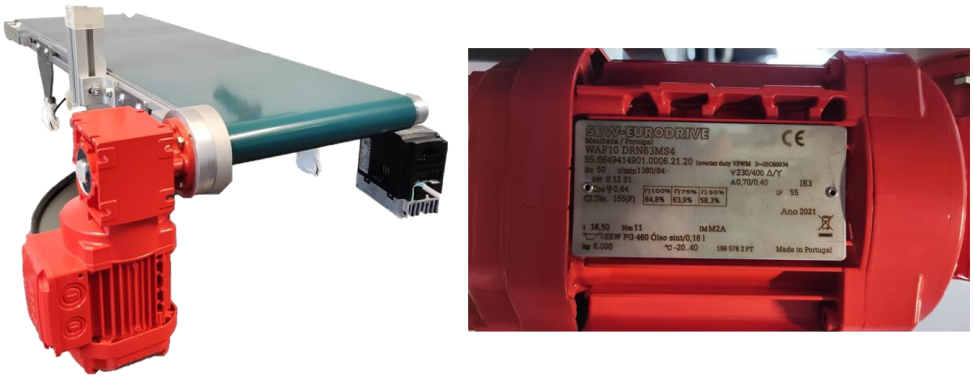


Figura 3.6 – Sistema de transporte, motor WAF10 DRN63MS4 de 0.12 kW.

A Figura 3.7 serve para demonstrar o esquema do protótipo sugerido com este projeto

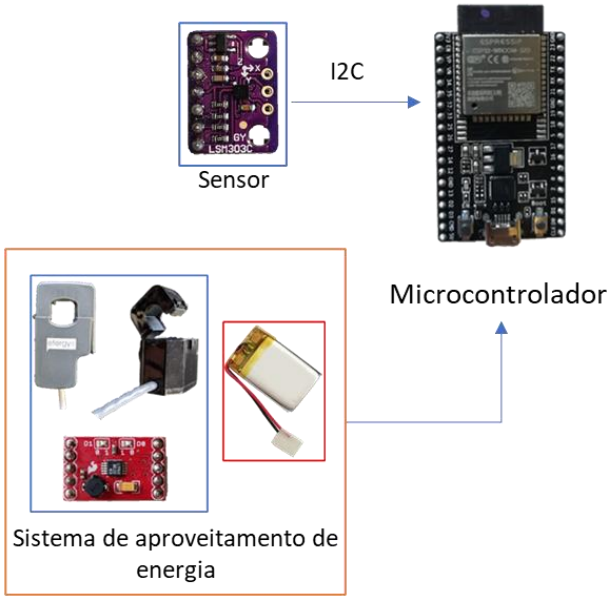


Figura 3.7 – Esquema do protótipo proposto.

3.2. Aquisição e algoritmo de classificação

No desenvolvimento e aquisição dos dados, a etapa dividiu-se em diferentes tarefas, sendo a primeira a escolha do melhor lugar para a aquisição, seguida pela criação do *dataset* e por fim a elaboração do sistema de classificação.

3.2.1. Criação do *dataset*

A aquisição dos dados foi feita, como já referido anteriormente, no motor trifásico visto que é um equipamento que conta com uma enorme presença na indústria. Numa primeira etapa, estudou-se a resposta em três lugares diferentes no motor para estudar qual o melhor local para captar aos valores de em relação aos três eixos tanto a nível do acelerómetro como do magnetómetro. Dois dos locais consistiam em apenas colocar o equipamento de aquisição acoplado em diferentes lugares (mais próximo e mais afastado do centro) do motor como se pode ver na Figura 3.8 – a) e b). No entanto, para garantir um bom acoplamento e estabilidade, o local escolhido foi posicionar o sistema na lateral do motor e para isso foi necessário criar um protótipo de acoplamento impresso em 3D (três dimensões) para fazer a interface entre o sistema e o motor, Figura 3.8 – c).

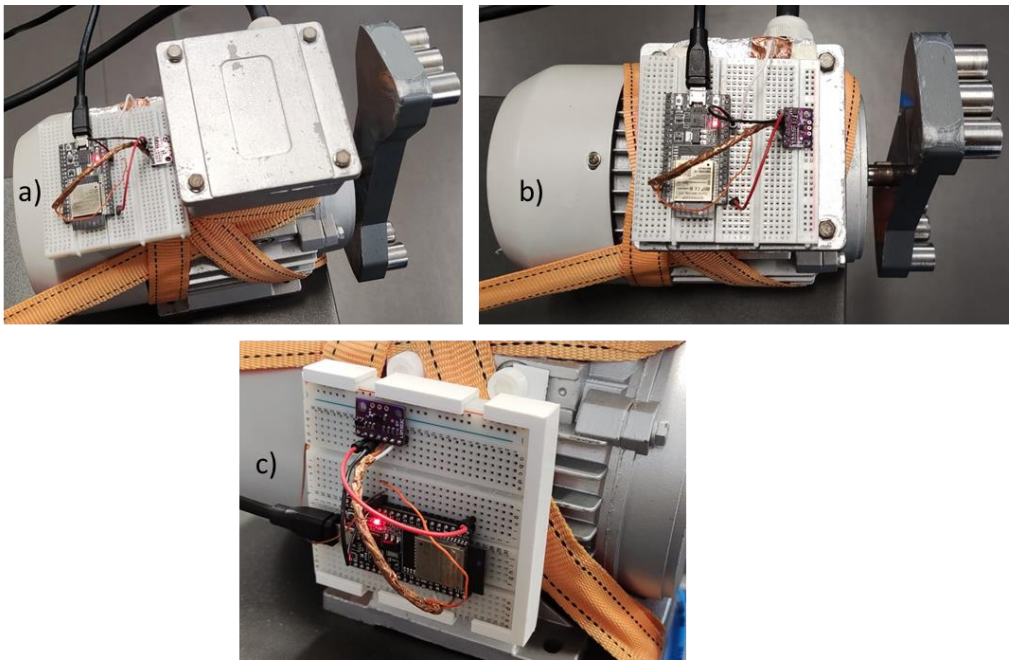


Figura 3.8 – Locais de estudo. a) Primeiro local, b) Segundo local, c) Terceiro local – lateral.

Após concluir o estudo do melhor local onde aplicar o sistema, estudo e resultados descritos na secção 4.1, procedeu-se à aquisição e criação do *dataset*. O *dataset* foi dividido em diferentes modos possíveis de funcionamento, e de maneira a simular as diferentes situações delineadas foi arquitetada e utilizada uma peça 3D para a acoplar ao motor, Figura 3.9, facilitando o uso e a distribuição de cargas (o conjunto de cargas é constituído por pesos em média de 86,8 gramas) de forma a simular anomalias (carga excêntrica) alternando a percentagem de pesos a utilizar em cada lado da peça, mas também sendo possível simular situações onde o funcionamento é feito em equilíbrio. As dimensões do acoplamento são: altura com 210mm (milímetros), a parte mais larga com 76mm e 40mm na parte mais estreita.

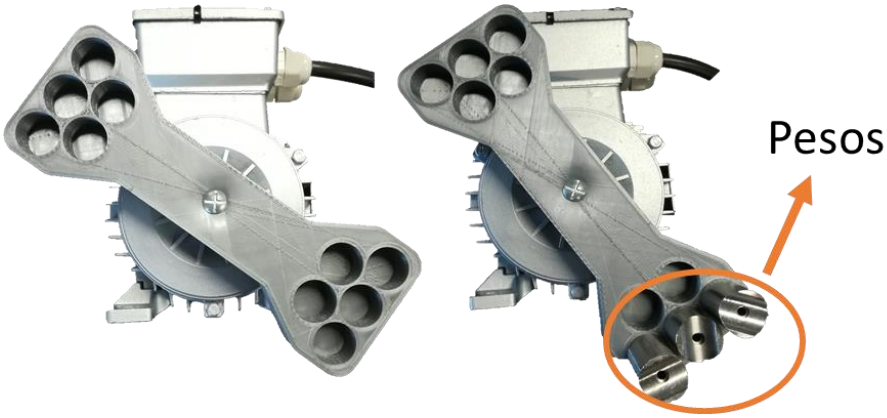


Figura 3.9 – Peça 3D acoplada ao motor juntamente com os pesos/cargas.

O *dataset* foi dividido em quatro modos de funcionamento principais:

Parado – O motor encontra-se parado tal como o nome indica.

Normal – Nesta situação o motor trabalha normalmente e sempre em equilíbrio.

Anomalia – Este modo é referente ao trabalho feito pelo motor em situação de desequilíbrio.

Anomalia Obstáculo – Por fim, este modo foi definido para situações semelhantes ao modo normal, no entanto conta com um obstáculo, simulado através de um batente, criando assim uma anomalia no seu funcionamento, Figura 3.10.

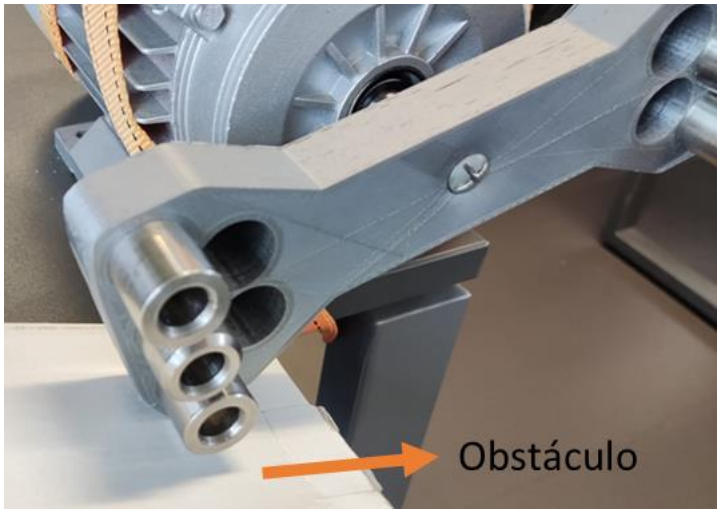


Figura 3.10 – Imagem representativa da "Anomalia Obstáculo", o obstáculo é visível a branco.

A aquisição foi realizada com o uso do *software Visual Studio Code* como ferramenta de edição juntamente com o *PlataformIO IDE* facilitando assim a interligação para a programação da placa de desenvolvimento ESP32. O *dataset* é constituído por aproximadamente trinta segundos de dados (aproximadamente 18750 linhas de dados) para todas as possíveis distribuições da carga pela peça 3D dentro das classes/modos funcionamento previamente definidas como é possível ver na Tabela 3.1 abaixo apresentada.

Tabela 3.1 – Modo de funcionamento e distribuição da carga.

Modo de funcionamento	Distribuição do nº de carga (Esq. – Dir.)									
	1 – 0	2 – 0	1 – 2	1 – 3	2 – 3	2 – 4	1 – 1	2 – 2	3 – 3	0 – 0 (Vazio)
Anomalia	X	X	X	X	X	X				
Normal							X	X	X	X
Parado										X
Anomalia Obstáculo							X	X	X	X

No caso de “Anomalia”, são retirados trinta segundos de informação referentes à situação onde é colocado uma só carga num dos lados, enquanto o lado oposto se encontra sem carga, após concluída essa aquisição passou-se para a distribuição de carga seguinte e assim adiante até todas as distribuições estarem adquiridas, o mesmo processo ocorreu para os restantes modos de funcionamento.

No estado “Normal” e no estado de “Anomalia Obstáculo” as leituras são sempre efetuadas numa condição de equilíbrio, iniciando-se numa situação de vazio, isto é, em ambos os lados da peça 3D não há presença de qualquer tipo de carga, nos cenários de distribuições seguintes aumentou-se de forma balanceada o número de carga presente na peça acoplada ao motor, Figura 3.11.

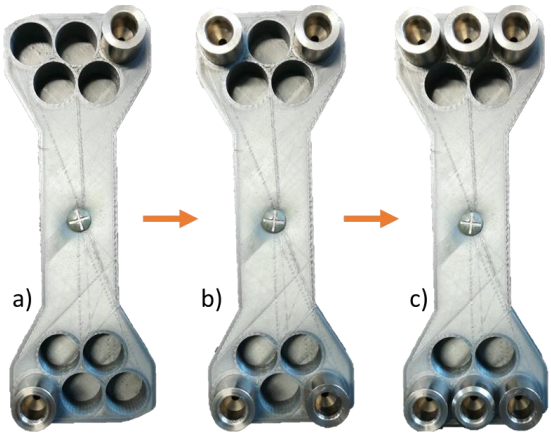


Figura 3.11 – Aumento da carga balanceada em modo “Normal” e modo “Anomalia Obstáculo”. a) Carga 1 – 1 b) Carga 2 – 2 c) Carga 3 – 3.

Por último, no modo “Parado” optou-se apenas por uma distribuição sem carga visto que nessa situação o motor não se encontra em funcionamento e dessa forma não sofre qualquer tipo de alteração referente há vibração, e em relação ao magnetómetro como o motor não é “alimentado” não existe um campo magnético girante em seu redor, o que em termos de leituras significa que nos valores retirados não se verifica variações.

Os valores foram adquiridos através do uso da porta série e gravados como ficheiros CSV (*Comma-separated values*), ficheiros que não contam com uma formatação especifica além de separados por virgulas, cada linha do ficheiro é constituída pela leitura dos dados no seu estado original pelo sensor LSM303C relativos ao acelerómetro e ao magnetómetro, ou seja, cada linha de informação é constituída por seis valores. Na Figura 3.12 é possível observar um exemplo da constituição de cada linha de informação obtida.

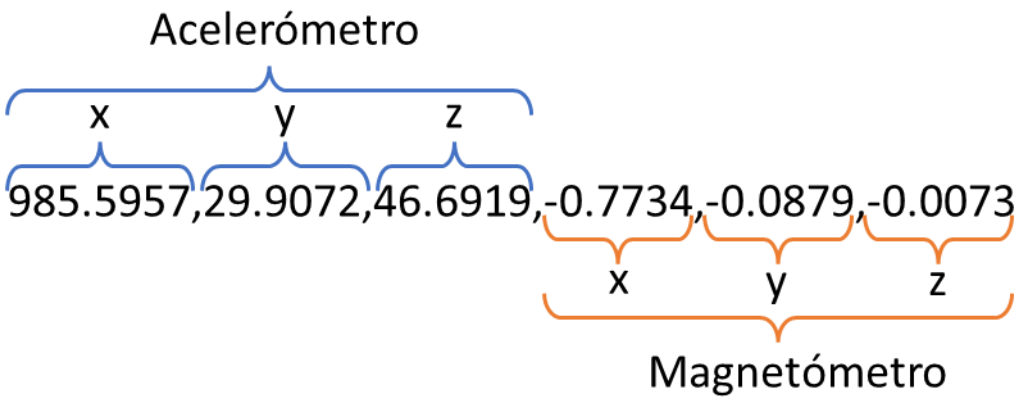


Figura 3.12 – Exemplo de uma linha de informação. A azul mostra os dados do acelerómetro e a laranja os dados do magnetómetro.

As configurações escolhidas do sensor para a aquisição dos dados foram, o protocolo de comunicação I2C para relativamente à sensibilidade do acelerómetro utilizou-se a escala de ±2g e para o magnetómetro utilizou-se a sensibilidade de ±16 gauss. Uma vez que neste projeto faz sentido o sensor obter novos valores/amostras o mais rapidamente possível, configurou-se o mesmo com as maiores taxas de dados a saída disponível sendo elas para o acelerómetro de 800Hz e para o magnetómetro de 80Hz. Por fim para a velocidade da comunicação serial utilizou-se o *baud rate* de 115200 bits por segundo.

Após recolher todos os dados de interesse e o *dataset* estar criado procedeu-se para a última etapa, a classificação dos dados, ou seja, a exploração de redes neuronais com capacidade de execução em microcontroladores.

3.2.2. Desenvolvimento da rede neuronal

O desenvolvimento do sistema de classificação, como foi referido anteriormente, efetuou-se após todo o *dataset* ter sido adquirido, no entanto antes de ser utilizado para a criação do modelo de classificação as seguintes etapas intermédias foram realizadas: 1) importação dos dados, 2) normalização e 3) divisão em treino e validação.

A ferramenta de escolha para a elaboração de todo este percurso foi a ferramenta online "*Google Colaboratory*", também conhecido por *Colab*, uma ferramenta de acesso gratuito que tem por base a técnica de *notebook* e a linguagem de edição na sua maioria é o "*python"*. Na Figura 3.13 é possível verificar as etapas e a sequencia seguida nesta parte do projeto.

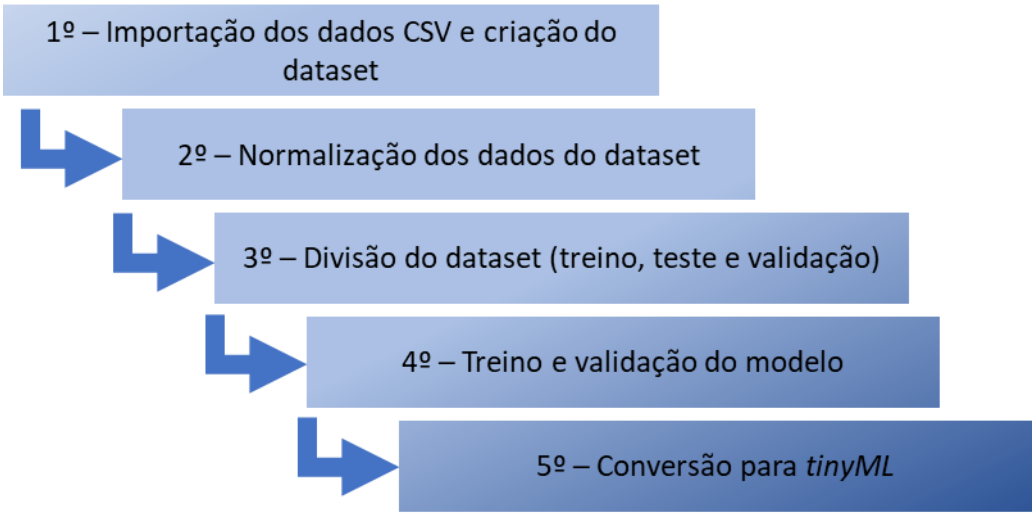


Figura 3.13 – Sequência de etapas efetuadas na ferramenta *Colab.*

O primeiro procedimento foi então a importação dos dados, de cada ficheiro são então retiradas 8000 linhas de informação que são divididos em grupos de 100 linhas/amostras, ou seja, para cada ficheiro de teste com uma distribuição da carga são associadas 100 amostras por 6 colunas (*features*) obtendo assim uma matriz [100*6].

De seguida, e para se obter dados mais coerentes efetuou-se o tratamento em relação aos valores do acelerómetro, visto o mesmo estar exposto à aceleração gravítica e dependente da orientação da instalação, a cada valor realizou-se uma subtração do valor médio da amostra em estudo, assim se o valor a tratar corresponde ao eixo X é lhe retirado o valor médio dessa mesma coluna na mesma amostra.

Por último procedeu-se a uma normalização dos dados, sendo a sensibilidade utilizada ±2g, dado que os valores lidos são em *mg* dividiu-se pelo maior valor possível (2000), resultando assim em valores normalizados entre [-1 1].

Ao concluir todo o tratamento do *dataset* procedeu-se à divisão do mesmo em dados de treino, dados de validação e dados de teste. Por dados de treino são os dados usados para treinar o modelo/rede com o intuito de dar a conhecer padrões com a respetiva classificação, quanto aos dados de validação os mesmos são utilizados para efetuar a validação durante a aprendizagem do modelo e por fim, os dados de teste servem para atestar a precisão e avaliar se o modelo criado funciona, estes dados são independentes do conjunto de treino e de validação. Assim sendo o *dataset* ficou então dividido em 60% para dados de treino, 20% para dados de validação e os restantes 20% para os elementos de teste.

A rede escolhida para o projeto foi uma rede CNN, devido a ser uma das opções para reconhecimento de padrões em imagens, mas também em sequências de dados ao longo do tempo, as primeiras camadas são correspondentes a convoluções 2D (duas dimensões), seguido de camadas para regularizar e por fim uma camada "*dense*" com ativação "*softmax*" obtendo-se assim as probabilidades para cada classe com valores entre 0 e 1, que correspondem aos principais modos de funcionamento. A estrutura da rede é visível no Anexo I, no entanto, os parâmetros de treino são apresentados de seguida na Tabela 3.2

Tabela 3.2 – Parâmetros de treino da rede.

Parâmetros	
Optimizador	Adam
Learning Rate	0,0001
Loss	*Sparse Categorical Crossentropy*
Epoch	1000 e 2000
Batch size	64

Os parâmetros da rede foram baseados inicialmente em [44], visto que utiliza valores e caraterísticas muito semelhantes, no entanto foram efetuados dois treinos diferentes, inicialmente com 2000 *epochs* e de seguida com 1000 *epochs*. No entanto, como a rede desenvolvida tem como intuito ser implementada no microcontrolador, ou seja, procedeu-se a uma conversão do modelo *tensorflow* para *tensorflow lite* e por fim para uma matriz de bytes tornando assim possível a sua aplicação no sistema embebido.

3.3. Sistema de monitorização e visualização

Uma vez concluídas as tarefas anteriores, e testada a precisão do modelo criado, procedeu-se ao desenvolvimento da solução final, capaz de recolher dados, analisar os mesmos e dando como resultado o seu modo de funcionamento. No entanto surgiu também a necessidade de visualização de todo o processo, desde os dados lidos e qual o modo de funcionamento, para isso foi necessário guardar toda a informação numa base de dados externa e por fim disponibilizar toda a informação numa interface gráfica, tornado assim a disposição dos dados de fácil leitura e compreensão. A Figura 3.14 apresenta uma visão geral sobre o desenvolvimento da solução final.

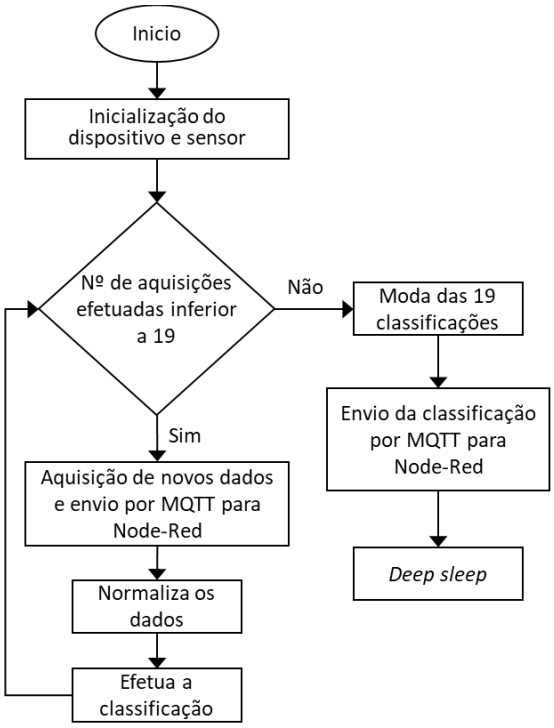


Figura 3.14 – Diagrama do desenvolvimento da solução final.

O desenvolvimento para a solução final dividiu-se em diferentes partes, numa fase inicial focou-se no desenvolvimento do sistema que fosse capaz de adquirir e analisar os dados, o sistema utiliza o modulo de WiFi no qual é ativado desde o início pois é necessário para a comunicação e envio dos dados, onde é explicado mais à frente.

De seguida inicia-se os parâmetros do sensor para se proceder a aquisição de 100 linhas de leituras que são enviadas para a interface gráfica, após completar essas leituras procede-se à normalização das mesmas de forma igual a que os dados de treino foram submetidos. Uma vez normalizados são inseridos na rede no qual é devolvido o resultado da classificação gravando o modo de funcionamento correspondente, após ser feita a análise procede-se a uma nova aquisição de valores repetindo o ciclo anteriormente descrito, no total são feitas 19 classificações, número esse para haver um valor ímpar de classificações e de assim não haver um empate entre duas classificações diferentes.

Ao fim de todas as classificações é aplicado a medida estatística da moda, ou seja, após as predições é analisado qual o modo de funcionamento com mais classificações, após o identificar o modo mais frequente procede-se ao envio para a disponibilizar a informação. O *broker* escolhido foi o "*Mosquitto*".

Como foi referido anteriormente os dados obtidos são processados e enviados, para isso acontecer recorre-se ao protocolo MQTT (*Message Queuing Telemetry Transport*), que através da ferramenta "Node-RED" é possível descodificação os dados recebidos remotamente, servindo como ponte entre o envio (do ESP32) para a base de dados *InfluxDB* onde são guardados e disponibilizados visualmente pelo *Grafana*, ferramenta de visualização interativa dos dados.

3.3.1. Protocolo de comunicação MQTT

O MQTT é um protocolo de dados utilizado para conectar dispositivos de IoT de baixa potência projetado para uma comunicação "*machine to machine*", o protocolo é executado com base em TCP/IP (*Transmission Control Protocol/Internet Protocol*). O protocolo MQTT foi normalizado pela ISSO/IEC 20922 que utiliza o uma metodologia de *publish/subscribe* como transmissão de mensagens o que resulta numa redução nos recursos necessários e torna este modelo ideal em situações onde existe uma largura de banda reduzida [45].

Num sistema MQTT são necessárias duas partes importantes, a primeira é o servidor conhecido como *broker* que server como intermediário entre os diversos clientes, evitando assim que os clientes não enviem mensagens diretamente uns para os outros. As mensagens contem um tópico para o qual os clientes podem publicar e subscrever, através do tópico o *broker* consegue gerir e enviar as mensagens certas para os clientes conectados e inscritos no tópico específico [46], [47].

Exites outros tipos de protocolos de comunicação como por exemplo o HTTP (*Hyper Text Transfer Protocol*) e o COAP (*Constrained Application Protocol*), o protocolo HTTP é um protocolo bem-conceituado para uso de uma comunicação de informação através da internet, no entanto não é utilizado em casos de IoT devido ao custo/complexidade associada, duração de bateria pois tem associado um enorme consumo energético. O HTTP usa o sistema de *request*/*response* como modelo de mensagem [47], [48].

O protocolo COAP é um protocolo baseado em HTTP virado para combater limitações do mesmo permitindo assim a implementação em sistemas com baixos recursos disponíveis, o COAP utiliza uma metodologia de *client/server*, este protocolo é utilizado para uma comunicação *machine to machine*. Ao contrário do MQTT que utiliza tópicos, o COAP utiliza URI (*Universal Resource Identifier*) [47], [48].

As diferenças entre o protocolo COAP e MQTT em termos de comunicação são: o COAP foca-se em ligações um para um, ou seja, transfere informações entre o cliente e o servidor enquanto o MQTT permite obter comunicação e enviar informações entre vários clientes. O protocolo MQTT consegue obter uma comunicação de baixo custo, uma grande flexibilidade no uso de sistemas IoT, e o baixo consumo em termos de energia, que é um aspeto importante devido à segunda parte do projeto, além da compatibilidade com as ferramentas de gestão escolhidas para este projeto, o Node-RED [49].

Através do protocolo MQTT são envidados os seguintes dados: 1) acelerómetro, 2) valor de temperatura e após efetuar as 19 classificações é enviado o 3) modo de funcionamento com as classificações.

3.3.2. Ferramenta de conexão Node-RED

O Node-RED é uma ferramenta de desenvolvimento baseada em fluxo, de código aberto, criada para implementar e conectar sistemas de IoT. Esta ferramenta permite conectar nós de entrada, saída e de processamento criando assim um fluxo de dados desde o sensor até à *cloud*. O Node-RED pode ser usado também para transferir dados para um nível superior, por exemplo uma base de dados, de forma instantânea [50].

Fornece ainda a possibilidade de criar uma interface gráfica, demonstrando ser uma ferramenta flexível e importante para a criação de protótipos de aplicações IoT de forma fácil e simples de maneira a configurar aplicações em tempos reais, graças a estas vantagens o uso do Node-RED é benéfico para a indústria [51].

Numa fase inicial esta ferramenta serviu para receber e processar os dados recebidos do sistema de monitorização por via MQTT e numa fase final serviu também como ferramenta de ligação com a base de dados *InfluxDB*.

Na Figura 3.15 é possível observar um exemplo de fluxo de código no editor do Node-RED, o mesmo é responsável de receber os valores enviados, e de proceder ao tratamento dos mesmos. Por fim encaminha os dados otimizados para a base de dados *InfluxDB*.

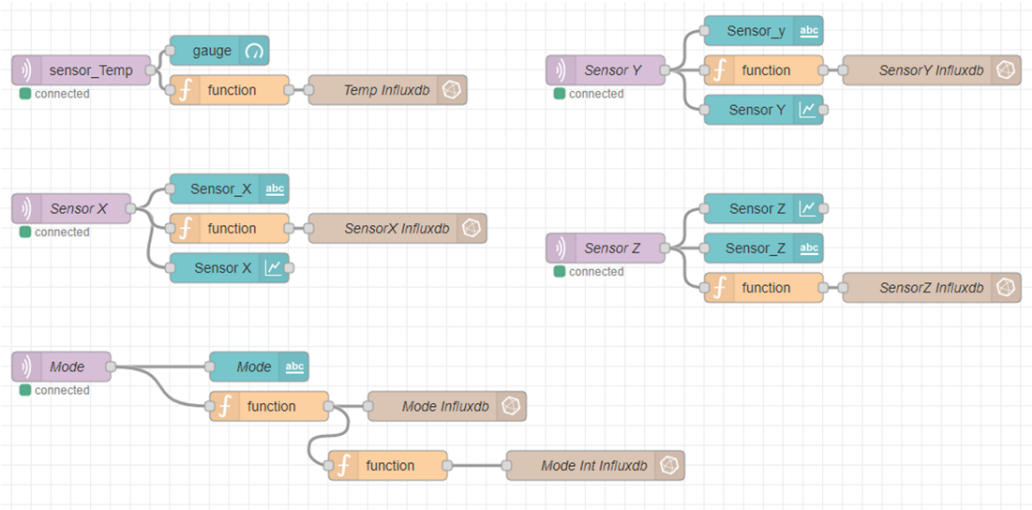


Figura 3.15 – Exemplo real do fluxo de código em Node-RED criado para o projeto.

3.3.3. Base de dados *InfluxDB* e ferramenta de visualização *Grafana*

O *InfluxDB* é um *time series database* (base de dados temporais) gratuita, uma *time series database* é uma base dados que tal como o nome indica é indicado para dados com uma componente temporal associada.

Esta base de dados não conta com dependências externas o que permite executar sem ser necessário instalar ou configurar, fornece o uso de uma linguagem semelhante a SQL de forma a aceder aos dados [52].

O principal benefício do *InfluxDB* é a capacidade de agrupar valores em tempo real através do uso de *buckets*, conta também com formas de gerir/purgar dados antigos, como por exemplo, ao fim de algum tempo elimina ou substitui por novos [53], [54].

O uso de uma ferramenta de *front-end* como o *Grafana* permite a visualização dos dados ao longo do tempo tornando eficaz a análise sobre os mesmos. O *Grafana* é uma plataforma gratuita que, como já foi referido, permite consultar, visualizar os dados temporais armazenados o que torna a sua utilização com o *InfluxDB* perfeita, uma vez que o seu uso também é recomendado pelos criadores do *InfluxDB* [52].

Na Figura 3.16 é possível observar a interface gráfica onde estão representados os dados retirados por parte do acelerómetro (retângulo vermelho), no retângulo amarelo é apresentado o modo de funcionamento do motor

tanto em forma de alerta como em forma de um gráfico. Por fim, são apresentadas informações acerta da temperatura captada pelo sistema, bem como o valor máximo, mínimo e o valor médio no intervalo de tempo selecionado e em forma de um gráfico é possível observar o comportamento da temperatura ao longo do tempo, região assinalada a verde.



Figura 3.16 – Interface gráfica, a vermelho a informação referente do acelerómetro, a amarelo as classificações efetuadas e a verde os dados da temperatura.

3.4. Estudo e implementação da técnica de *energy harvesting*

Após concluir as etapas anteriores, prosseguiu-se para o desenvolvimento e o estudo da parte final deste projeto, sendo ela a captação de energia em redor do motor selecionado. Na Figura 3.17 é possível verificar as etapas efetuadas no recorrer desta parte do projeto.

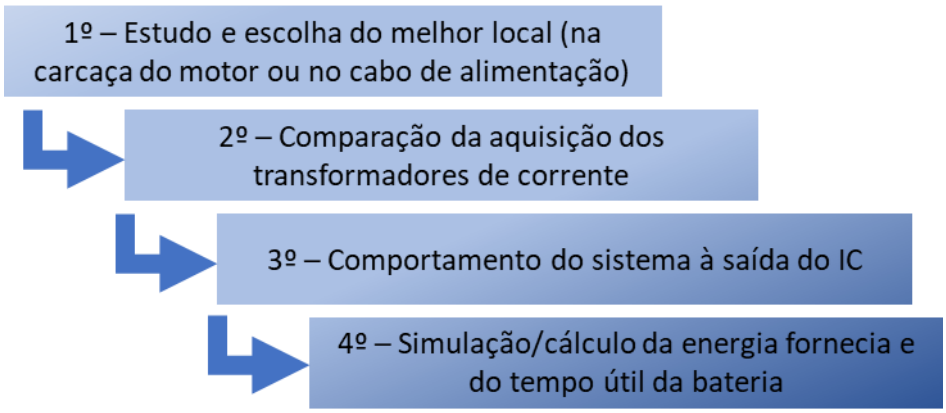


Figura 3.17 – Etapas efetuadas no estudo do aproveitamento de energia.

Inicialmente estudou-se os principais locais onde colocar os transformadores de corrente. Para o estudo escolheu-se locais estratégicos, no qual se adquiriu o sinal dos transformadores de corrente a diferentes frequências de funcionamento do motor, com o intuito a escolher o local e o sensor mais indicado (melhor

performance). Os locais em escolha foram em torno do motor com o intuito de captar os campos magnéticos, mas também acoplando o sensor aos cabos de alimentação do motor, na Figura 3.18 é possível observar os locais estratégicos escolhidos para a aquisição.

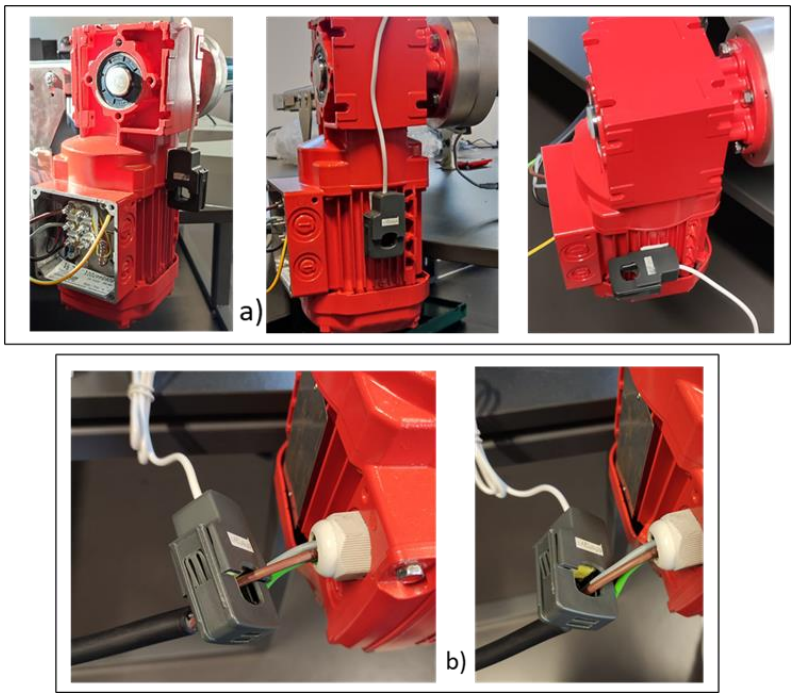


Figura 3.18 – Locais de adquirição de energia. a) Posições ao redor do motor b) Acoplamento nos cabos de alimentação.

Uma vez escolhido o melhor local no qual se aproveita melhor energia, estudos e resultados descritos na secção 4.2, procedeu-se para a escolha do transformador com o melhor resultado, e recorrendo à ajuda do circuito integrado LTC3588, referido anteriormente, e inserido no protótipo criado para a recolha e aproveitamento da energia, protótipo esse visível na Figura 3.19 devidamente dividida.

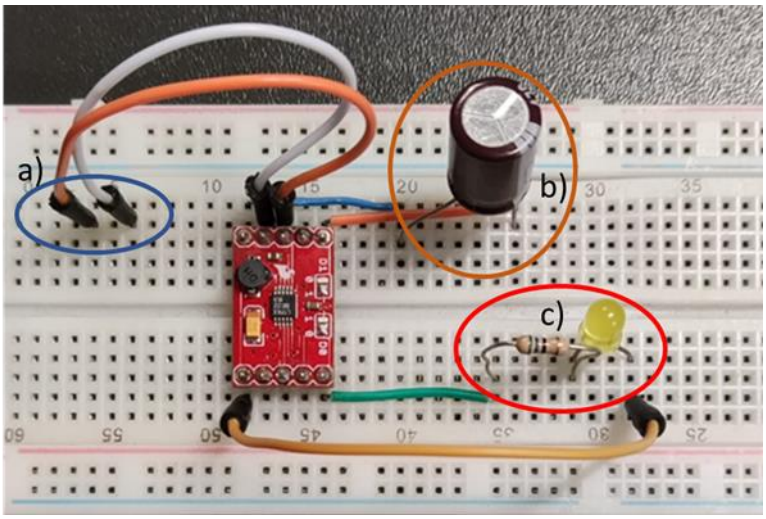


Figura 3.19 – Protótipo de *energy harvesting*. a) Entrada para o IC b) Armazenamento de energia c) Saída do IC.

Foram efetuadas diferentes análises à saída do protótipo do funcionamento dos dois transformadores de corrente, num primeiro caso estudou-se o tempo de carregamento desde o início do primeiro armazenamento até à primeira disponibilização de tensão, ou seja, desde o armazenamento no condensador ser nulo até atingir a carga necessária no qual o circuito disponibiliza a tensão para a saída. Num segundo caso foi ainda analisado o tempo entre picos de tensão, ou seja, o tempo que demora a carregar novamente até gerar uma nova ativação da saída e para terminar estudou-se a duração de um pico servindo assim para calcular a energia gerada e dessa forma simular o tempo de vida útil.

Para simular a uma carga utilizou-se um LED que quando a tensão é disponibilizada à saída, o mesmo acende consumindo assim tensão armazenada. De forma a controlar as quantidades de tensão libertada, o circuito integrado apresenta um parâmetro responsável por esse mesmo controlo, disponibilizando à saída quando o armazenamento contém aproximadamente 5V e corta o acesso quando a tensão presente no condensador atinge valores abaixo dos 3.7V, procedendo novamente ao armazenamento de tensão/corrente formando assim um ciclo.

4. TESTES E RESULTADOS

Esta secção corresponde a apresentação dos testes efetuados ao longo de todo o projeto, bem como os resultados obtidos. Este capítulo divide-se em duas partes, na primeira parte descreve-se as experiências realizadas de forma direta ou indireta para a deteção de anomalias.

A segunda parte detalha os testes em relação ao estudo de utilização do *energy harvesting*, bem como o cálculo de energia gerado e a duração de prevista para todo o sistema.

4.1. Ensaio relacionado com a deteção de anomalias

Ao longo do desenvolvimento deste projeto, o primeiro grande desafio foi descobrir qual o melhor local onde implementar o sistema a desenvolver. Tal como foi referido anteriormente, foram efetuadas medições em três locais diferentes, comparando os valores de acelerómetro e de magnetómetro. Assim o local a) foi o primeiro a ser analisado, escolhido devido a situar-se perto de ventilação do motor e simultaneamente próximo do centro do motor, a posição b) foi escolhida por ser na caixa de ligação, ou seja, o local muito acessível onde se encontram os contactos de alimentação e de ligação do motor.

Deste modo na Figura 4.1 e na Figura 4.2 é possível observar as leituras obtidas nas duas posições a) e b) respetivamente, devidamente assinaladas, o gráfico do lado esquerdo corresponde ao acelerómetro, e do lado direito ao magnetómetro.

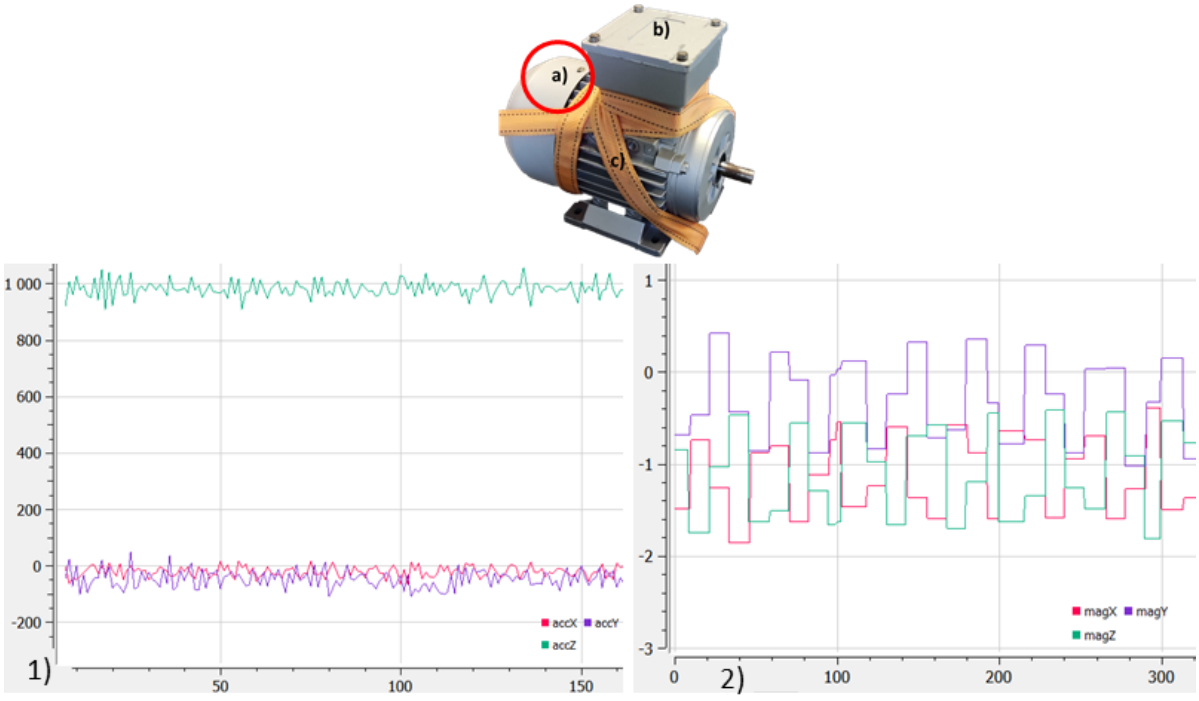


Figura 4.1 – Leitura do sensor na posição a). 1 – Acelerómetro: Eixo X – Nº amostras, Y – mg. 2 – Magnetómetro: Eixo X – Nº amostras, Y – G (Gauss).

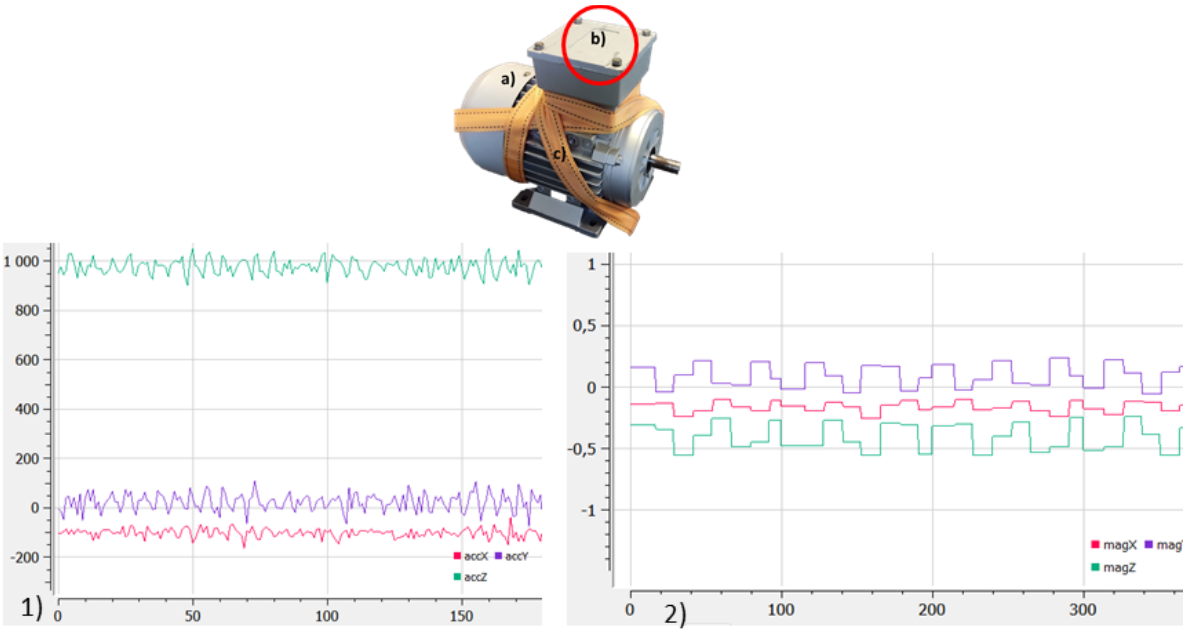


Figura 4.2 – Leitura do sensor na posição b). 1 – Acelerómetro: Eixo X – Nº amostras, Y – mg. 2 – Magnetómetro: Eixo X – Nº amostras, Y – G (Gauss).

Analisando-se os gráficos, do lado esquerdo demonstram os dados do acelerómetro numa escala de milésimos de G (força gravítica), o eixo Z do acelerómetro têm a mesma direção comparativamente ao eixo da gravidade e por esse motivo a oscilação verificada tem valor médio de 1000 que corresponde a 1G. Entre as duas posições a) e b), no que respeita ao acelerómetro, as diferenças existentes são muito pequenas e irrelevantes.

Os gráficos da direita são referentes aos valores obtidos do magnetómetro, leituras feitas em "Gauss" (unidade medição do fluxo magnético). Comparando as duas medições é possível observar diferenças na amplitude das oscilações das medições sendo que na posição a) estas apresentam-se com valores superiores, situação esperada devido ao local de aquisição ser mais próximo do estator do motor trifásico o que permite a captação de campos magnéticos mais intensos. Na posição b) isto não se verifica visto que se encontra numa posição mais distante da carcaça do motor.

Por último realizou-se aquisição na posição c), posição lateral nas ranhuras presentes na carcaça do motor, colocando deste modo o sistema orientado dentro do campo girante existente no equipamento. Assim, a Figura 4.3 representa os dados obtidos sendo que o gráfico do lado esquerdo corresponde ao acelerómetro, e do lado direito ao magnetómetro.

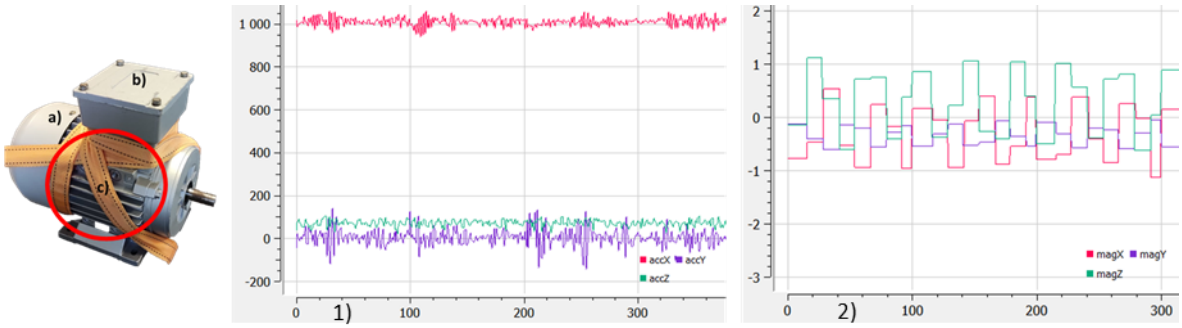


Figura 4.3 – Leitura do sensor na posição c). 1 – Acelerómetro: Eixo X – Nº amostras, Y – mg, 2 – Magnetómetro: Eixo X – Nº amostras, Y – G (Gauss).

Pelo gráfico dos dados do acelerómetro, é possível verificar um aumento nas variações e oscilações detetadas pelo sensor, o que demostra que através do processo de acoplamento mecânico do sistema, este deteta mais facilmente as vibrações do motor. Em relação aos valores do campo magnético na posição c) e como se situa perto do campo girante os valores obtidos foram mais altos comparativamente com as posições testadas anteriormente, o que significa que presença do campo magnético nesta posição é maior tal como esperado.

Analisando de uma forma geral os resultados obtidos nas três posições, no que respeita ao acelerómetro as diferenças não são muito notórias, no entanto na posição c) devido ao sistema ficar acoplado rigidamente ao motor torna a aquisição de dados mais eficaz comparativamente às outras posições uma vez que em ambas o sistema fica apenas com um contacto não rígido.

As maiores diferenças são visíveis ao nível da leitura do campo magnético uma vez que todas as posições se encontram em locais onde a intensidade do campo magnético é diferente. Porem, é no local c) que se adquire um valor mais elevado. Assim, em consideração a todos os resultados, o local escolhido para a colocação do sistema foi o local c). A Figura 4.4 demonstra exemplos de dados adquiridos sobre os modos de funcionamento, acerca do acelerómetro uma vez que nem relação ao magnetómetro apenas a intensidade muda.

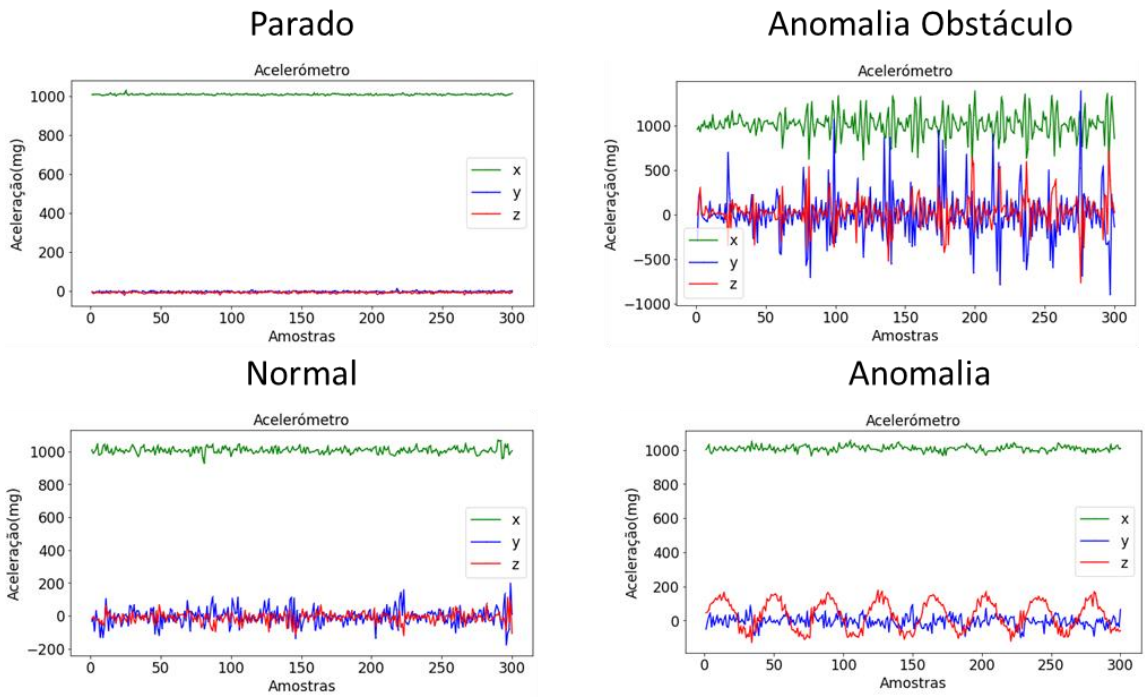


Figura 4.4 – Dados exemplos do acelerómetro nos diferentes modos de funcionamento.

Após o estudo e aquisição estarem completos, como se referiu anteriormente, procedeu-se à criação da rede neuronal (ver estrutura no Anexo I) para classificação dos dados, deste modo foram testadas duas situações diferentes em relação ao parâmetro das *epochs*. No primeiro caso o treino foi efetuado com 2000 *epochs* com o intuito de analisar o comportamento do mesmo em relação à evolução das métricas de "*loss"* e de "*accuracy"* para obter uma visão da evolução do treino. A Figura 4.5 representa o comportamento das métricas ao longo das *epochs*.

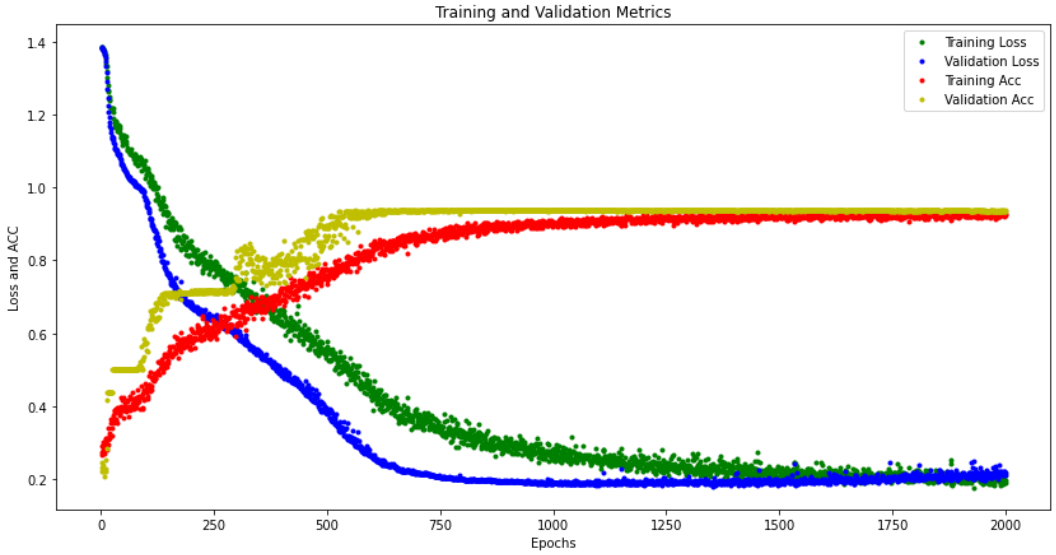


Figura 4.5 – Evolução da *loss* e da *accuracy* ao longo do treino de 2000 *epochs*.

Realizando uma análise à evolução do treino ao longo das *epochs* é notório que ambas as métricas estabilizam na sua validação por volta da *epoch* número 800. Posteriormente, com a conclusão do treino e com a ajuda dos dados de teste obtidos com a anterior divisão do *dataset*, verificou-se a *accuracy* do modelo obtido. Desta forma na Figura 4.6 é possível observar o valor, sendo ele de 90.42%.

```
8/8 [==============================] - 0s 10ms/step - loss: 0.2079 - accuracy: 0.9042
Restored model, accuracy: 90.42%
```

Figura 4.6 – Valor da *accuracy* com dados divididos para teste.

Com a finalidade de descobrir o desempenho e a performance, obteve-se a matriz de confusão sobre do modelo com o intuito de observar a distribuição das classificações. Esta consiste em quatro caraterísticas usadas para definir as métricas do classificador, sendo elas:

***True Positive* (TP) –** Representa o número de classificações positivas devidamente classificadas;

***True Negative* (TN) –** Representa o número de classificações negativas classificadas com precisão;

***False Positive* (FP) –** Significa o número de classificações negativas que foram classificadas como positivas;

***False Negative* (FN) –** Representa o inverso do "FP", ou seja, significa os valores positivos que foram classificados como negativos.

Na Figura 4.7 é possível observar a matriz de confusão obtida com os dados de teste, neste caso o modelo consiste em quatro classes, ou seja, uma classificação multiclasse, sendo possível verificar a percentagem de precisão de classificação de cada modo de funcionamento, os valores da precisão variam entre 0 e 1.

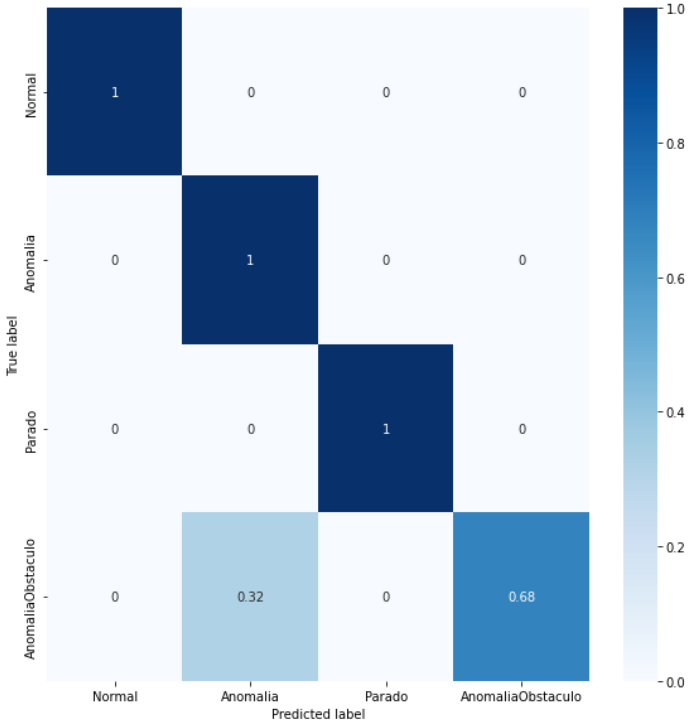


Figura 4.7 – Matriz de confusão sobre os dados de teste, com 2000 *epochs*.

Com a análise da matriz de confusão obtida, é possível observar que a classificação com os dados do *dataset* de teste em relação ao modo de funcionamento "Normal", "Anomalia" e "Parado" obteve uma *accuracy* de 100%, em contrapartida no modo "Anomalia Obstáculo" apenas acerta 68% classificando os restantes 32% como "Anomalia".

Como já foi referido anteriormente, o treino do modelo de 2000 *epochs* estabilizou aproximadamente em torno da *epoch* 800, uma vez que se verificou não ser necessário um número elevado de *epochs* efetuou-se um treino do modelo com a metade referente ao treino anterior, ou seja, com 1000 *epochs*, no entanto mantendo os restantes parâmetros com os mesmos valores. Na Figura 4.8 é visível o comportamento das métricas da *loss* e da *accuracy* quanto ao *dataset* de treino e do *dataset* de validação.

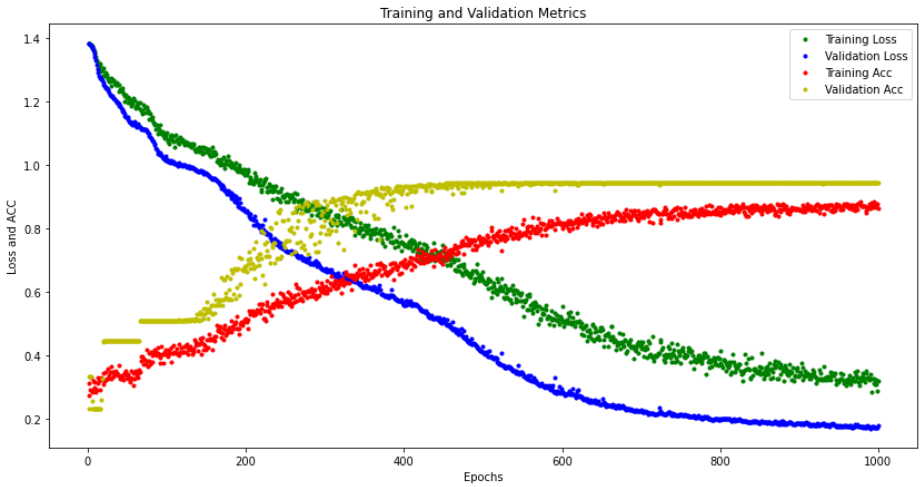


Figura 4.8 – Evolução da *loss* e da *accuracy* ao longo do treino de 1000 *epochs*.

Efetuando uma análise à evolução do treino ao longo das *epochs* é visível que a *accuracy* de validação estabiliza por volta da *epoch* número 600 em relação as restantes métricas, ambas estabilizam por volta da *epoch* número 900, no entanto não é visível uma alteração significativa na *accuracy* de validação do treino.

Após concluir o treino e tal como no anterior, recorrendo aos dados de teste obtidos com a divisão do *dataset*, verificou-se a *accuracy* do modelo obtido, na Figura 4.9 é possível observar o valor percentual de acerto utilizando os dados de teste, sendo ele de 92.50%. Por fim, e de forma a terminar o estudo/criação do modelo obteve-se a matriz de confusão observando o desempenho e a performance do modelo, na Figura 4.10 é possível observar a matriz obtida.

```
8/8 [==============================] - 0s 5ms/step - loss: 0.1811 - accuracy: 0.9250
Restored model, accuracy: 92.50%
```

Figura 4.9 – Valor de *accuracy* utilizando os dados de teste.

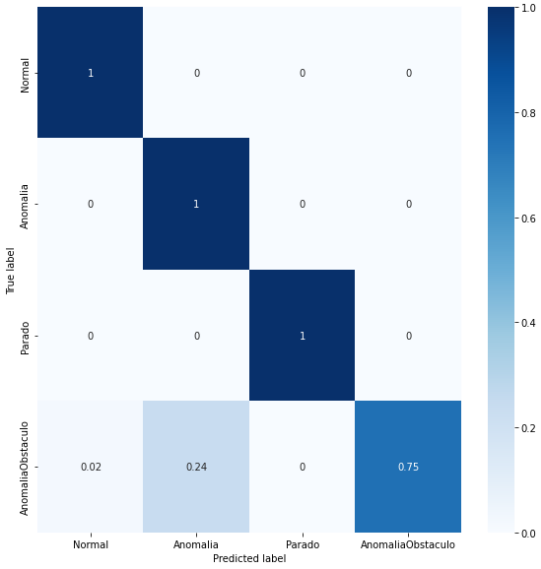


Figura 4.10 – Matriz de confusão dos *dataset* de teste do modelo com 1000 *epochs*.

Com a análise da matriz de confusão anterior, é possível observar que novamente a classificação com os dados do *dataset* de teste em relação ao modo de funcionamento "Normal", "Anomalia" e "Parado" obteve uma *accuracy* de 100%, mas desta vez no caso modo "Anomalia Obstáculo" apenas é correto em 75% das vezes, classificando os restantes 25% maioritariamente como "Anomalia", no entanto 2% desses 25% tem como resultado de classificação o modo de funcionamento "Normal".

O modelo escolhido para ser utilizado na placa de desenvolvimento ESP32 foi o modelo treinado com 1000 *epochs*. uma vez ter apresentado melhor desempenho no *dataset* de treino.

Após converter o modelo para *tensorflow lite* e ter sido incorporado no microcontrolador procedeu-se a fase de teste e de validação do mesmo. O processo é composto pela análise do motor em trabalho para todas as distribuições de carga referentes aos modos de funcionamento, no qual foram efetuados 10 testes para cada distribuição. A Figura 4.11 representa o código utilizado para converter o modelo para *tensorflow lite* para que seja possível implementar no microcontrolador, código esse desenvolvido com base no exemplo fornecido pela "Tensorflow".

```
# Definir paths para os ficheiros do modelo
import os
MODELS_DIR = 'models/'
if not os.path.exists(MODELS_DIR):
    os.mkdir(MODELS_DIR)
MODEL_TF = MODELS_DIR + 'model'
MODEL_NO_QUANT_TFLITE = MODELS_DIR + 'model_no_quant.tflite'
MODEL_TFLITE = MODELS_DIR + 'model.tflite'
MODEL_TFLITE_MICRO = MODELS_DIR + 'model.cc'

# Converter o modelo para formato TensorFlow Lite
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

# Guardar o modelo no disco
open(MODEL_TFLITE, "wb").write(tflite_model)

basic_model_size = os.path.getsize(MODEL_TFLITE)
print("Model is %d bytes" % basic_model_size)

# Instalar xxd
!apt-get update && apt-get -qq install xxd
# Converter para ficheiro C, um modelo TensorFlow Lite para microcontroladores
!xxd -i {MODEL_TFLITE} > {MODEL_TFLITE_MICRO}

# Update variaveis
REPLACE_TEXT = MODEL_TFLITE.replace('/', '_').replace('.', '_')
!sed -i 's/'{REPLACE_TEXT}'/g_model/g' {MODEL_TFLITE_MICRO}
```

Figura 4.11 – Código utilizado para a conversão do modelo em *tensorflow lite*.

Cada teste consiste em garantir a obtenção de 19 classificações distintas sobre o modo de funcionamento do motor (com amostras de 100 valores), observando a distribuição das mesmas pelas diferentes classes de funcionamento/modos. A Tabela 4.1 serve como demostração e exemplo representativo, no qual estão apresentados resultados de três testes, no Anexo II é possível analisar os restantes resultados.

No total serão 190 classificações para cada distribuição de cargas/pesos, o modo "Anomalia" conta seis diferentes distribuições, o que no total resulta de 1140 classificações, no modo "Normal" e "Anomalia Obstáculo" totalizam quatro distribuições resultando em 760 classificações para cada modo e por fim existe apenas uma distribuição sobre do modo "Parado".

Tabela 4.1 – Resultado e distribuição das classificações.

Modo	Distribuição carga	Teste nº 1	Teste nº 2	Teste nº 3
		Nº ocorrências por classe [Normal],[Anomalia],[Parado],[Anomalia Obstáculo]		
Anomalia	1 – 0	[0],[19],[0],[0]	[0],[19],[0],[0]	[8],[11],[0],[0]
	2 – 0	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]
	1 – 2	[0],[19],[0],[0]	[0],[19],[0],[0]	[1],[18],[0],[0]
	1 – 3	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]
	2 – 3	[1],[18],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]
	2 – 4	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]

Como já foi referido, na tabela é possível observar as classificações sobre a classe “Anomalia” e com os resultados obtidos é visível que classifica em quase todas as distribuições as 19 amostras como “Anomalia”, havendo apenas um teste referente a distribuição de “1 – 0” no qual apresentou mais dificuldade.

No total há 190 classificações para cada distribuição de carga, o modo “Anomalia” contando com seis maneiras diferentes de distribuições o que no total resulta de 1140 classificações, no modo “Normal” e “Anomalia Obstáculo” que conta com quatro distribuições resultando em 760 classificações para cada modo e por fim existe apenas uma distribuição atribuída ao modo “Parado”.

Após a obtenção de todos os resultados, foi gerada uma matriz de confusão de forma a simplificar e apresentar as classificações, a Figura 4.12 apresenta todas as classificações referidas anteriormente.

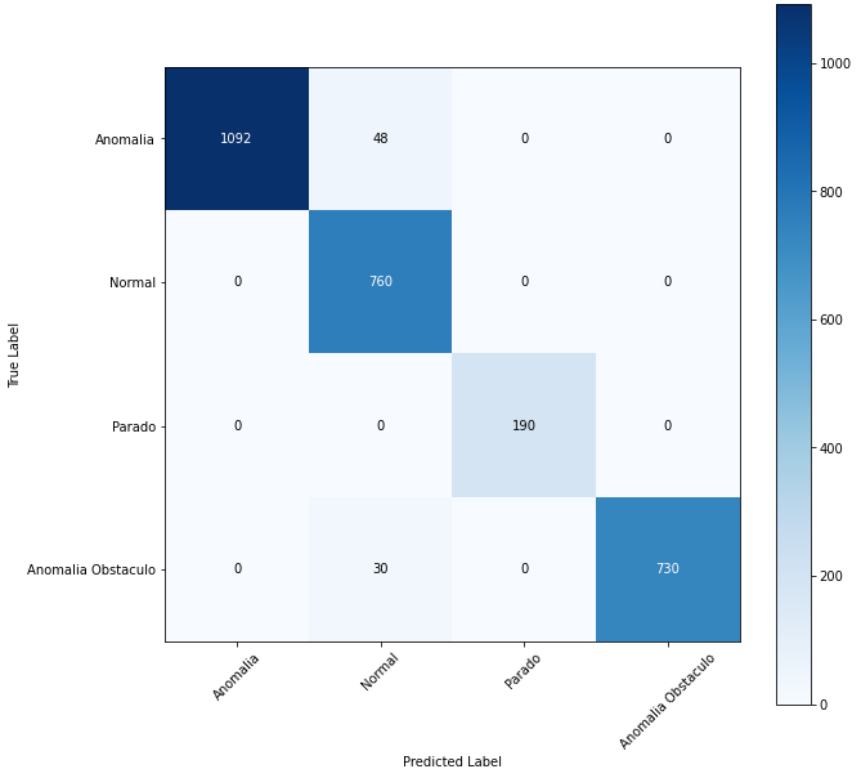


Figura 4.12 – Matriz confusão da distribuição das classificações.

As classificações referentes a classe “Anomalia”, 1092 das mesmas classificações retornaram o resultado esperado, no entanto as restantes 48 classificações retornaram modo “Normal”. Sobre as classificações dos modos de funcionamento “Normal” e o modo “Parado”, todas apresentaram elevada precisão, ou seja, em ambos os casos o modelo acertou em todos os testes. Por último, no caso do modo “Anomalia Obstáculo” apenas 30 dos testes não obtiveram a classificação devida, tendo sido rotuladas como modo “Normal”, no entanto, os restantes testes retornaram o resultado pretendido.

Num total de 2850 classificações houve apenas 78 classificações no qual o modelo não conseguiu classificar o modo correto de funcionamento. Na Figura 4.13 é visível a matriz de confusão com a representação da percentagem de precisão referente a cada modelo.

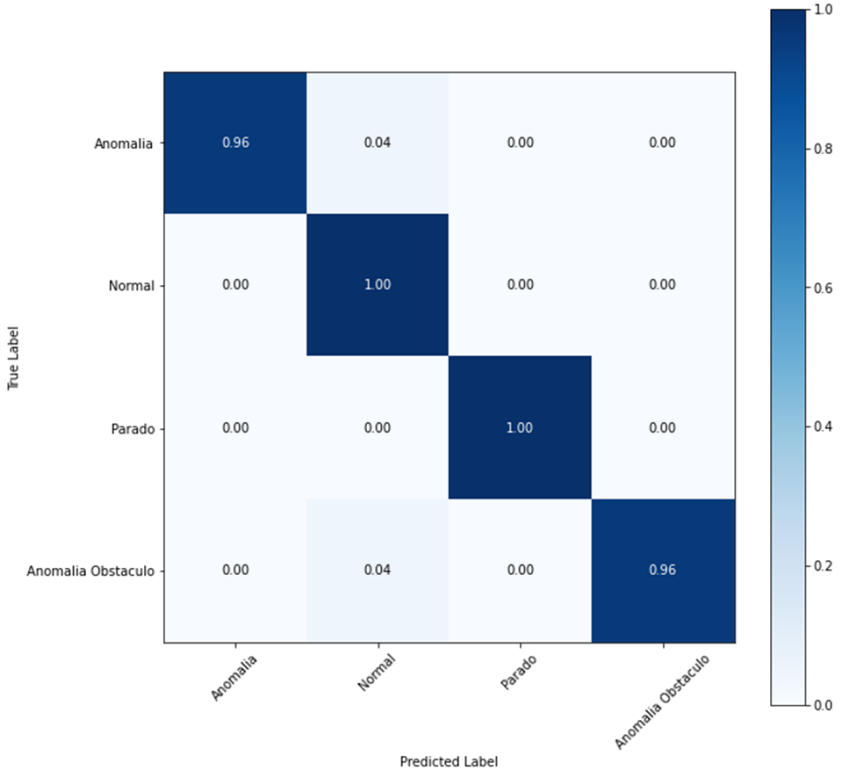


Figura 4.13 – Matriz de confusão final percentual.

A precisão final modelo foi de aproximadamente 97.26 %, sendo calculada através da divisão entre o número total de classificações corretas pelo número total de classificações efetuadas, ver equação 2.

$$Accuracy = \frac{Nº\ total\ de\ classificações\ corretas}{Nº\ total\ de\ classificações} * 100 \qquad (Eq.\ 2)$$

De forma a terminar esta etapa verificou-se o tempo que demora a fazer uma análise desde aquisição dos novos valores até entrar em *deep sleep*, sendo esse tempo de aproximadamente 13 segundos.

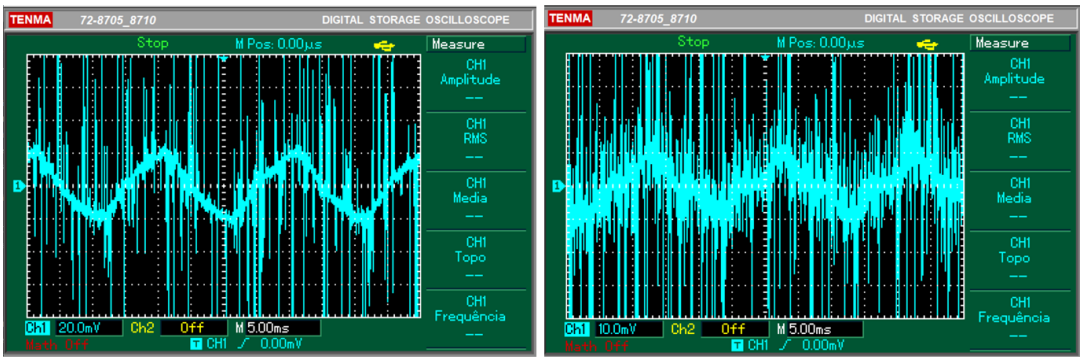
4.2. Ensaio da proposta de *energy harvesting*

Nesta secção são apresentados os testes e o estudo sobre da segunda parte do projeto, desde a aquisição do sinal por parte dos transformadores de corrente até ao estudo da melhor localização para extração de energia, complementada com a comparação entre dos dois modelos de transformador em teste.

4.2.1. Estudo do local de aquisição.

Inicialmente a análise de aquisição por parte dos transformadores de corrente ocorreu nas posições referidas na secção 3.4.

O primeiro estudo avaliou as diferentes posições de aquisição na carcaça do motor em funcionamento a 50Hz, no entanto os resultados não foram os mais favoráveis. Neste caso, o transformador cinzento (razão de transformação de 1:1350) obteve uma amplitude de onda bastante reduzida (na gama dos 30mV) como é visível na Figura 4.14 – a), a seguir apresentada. Com a utilização do transformador preto (razão de transformação de 1:2000) não se conseguiu interpretar o resultado obtido uma vez que o sinal de onda se encontrava com excesso de ruido, representado na Figura 4.14 – b). É de salientar que as formas de onda são adquiridas através do uso de um osciloscópio, Tenma 72-8710.



a) b)

Figura 4.14 – Aquisição dos transformadores de corrente. a) Transformador cinzento b) Transformador preto.

A mesma aquisição foi ainda efetuada para outras frequências de trabalho, sendo elas 10Hz, 20Hz, 30Hz e 40Hz além dos 50Hz referidos anteriormente, no entanto os resultados obtidos nas posições envolventes não sofreram alterações, ou seja, em relação ao transformador cinzento a amplitude de onda continuava reduzida e para o transformador preto o ruído mantinha-se.

De seguida testou-se a posição onde os transformadores ficam presos nos cabos fase de alimentação, de forma similar aos testes anteriores, foram analisadas em diferentes frequências de trabalho, desde 10 Hz até 50Hz, observando uma diminuição significativa no ruido do sinal adquirido. A figura seguinte demostra os resultados obtidos demostrando a onda adquirida pelos transformadores de corrente às frequências de 10Hz (Figura 4.15 – a)), 30Hz (Figura 4.15 – b)) e 50Hz (Figura 4.15 – c)), as três frequências servem assim para observar a evolução e as diferenças que as frequências de trabalho do motor provocam.

Na Figura 4.15 a escala correspondente ao valor eixo Y é mantida nos 200mV no entanto e devido à variação da frequência para os testes efetuados, a escala do eixo X variou-se de forma a permitir a visualização da forma de onda. Os sinais posicionados no lado esquerdo da figura correspondem ao transformador cinzento e os sinais do lado direito dizem respeito ao transformador de corrente preto.

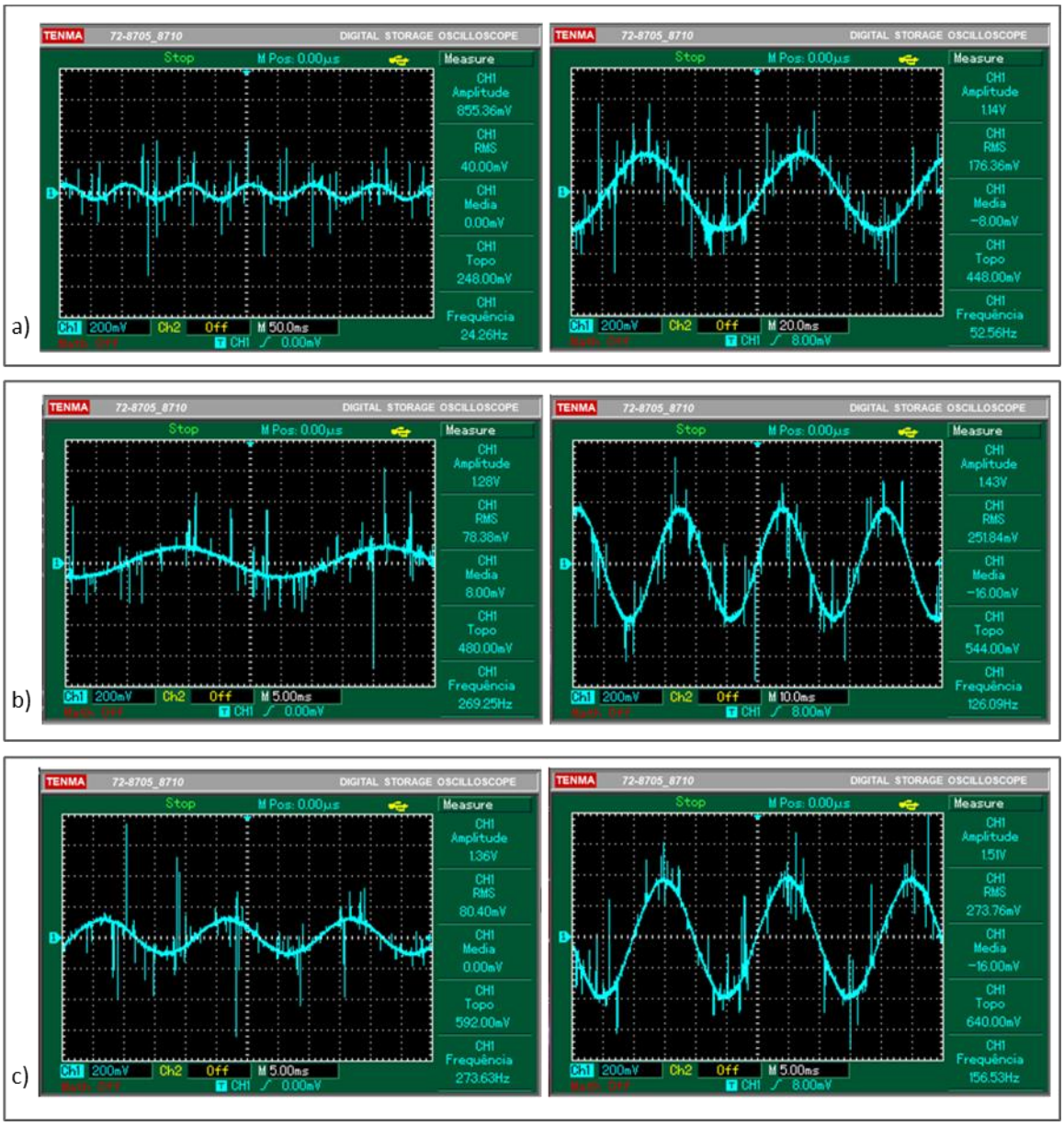


Figura 4.15 – Resultados dos transformadores de corrente posicionados em uma fase de alimentação. a) – Frequência de 10Hz; b) Frequência de 30 Hz; c) Frequência 50Hz.

Analisando os resultados obtidos a um nível geral, é visível que com o aumento da frequência de trabalho a amplitude do sinal adquirido também aumenta e a diferença entre os transformadores também é notória. Fazendo uma análise mais aprofundada à utilização do transformador cinzento, na frequência dos 10Hz obteve perto de 40mV de amplitude, nos 30Hz o sinal chega aos 120mV e por fim nos 50Hz o valor aproxima-se dos 160mV, no entanto comparando esses resultados com os obtidos ao utilizar o transformador de corrente preto, esses valores são baixos.

Num funcionamento a 10Hz o transformador de corrente com uma razão de transformação de 1:2000 é capaz de obter uma amplitude de 240mV, o que é um valor mais favorável relativamente aos resultados da

utilização do transformador anterior. O resultado na frequência seguinte, de 30Hz, tem um valor adquirido que ronda os 350-360mV e por fim no teste à frequência de 50Hz o sinal de onda atinge uma amplitude de sinal de 400mV, o que é um valor bastante significativo visto que o assunto em questão passa pelo reaproveitamento de energia disponível. Através dos resultados dá para perceber que o uso do transformador de corrente preto é o mais indicado.

Outro teste realizado de forma a comparar mais uma vez os dois sensores, foi o comportamento da saída na situação de placa de aproveitamento LTC3588 ter como fonte de alimentação os dois modelos de transformador. Os testes efetuados basearam-se maioritariamente sobre o tempo, como se já referiu anteriormente, numa primeira instância analisou-se o tempo que leva desde o primeiro momento em que se coloca o motor a funcionar até que acontece o primeiro "disparo" de saída, ou seja, o tempo que leva a carregar o condensador de 0V de carga até atingir a carga necessária (5V) que permita ativar a saída regulada.

Na Figura 4.16 é possível verificar o tempo que demora a carregar o condensador pela primeira vez, o gráfico a) é referente ao uso do transformador de corrente cinzento e o segundo gráfico b) diz respeito ao transformador preto. Os intervalos de tempo estão devidamente assinalados.

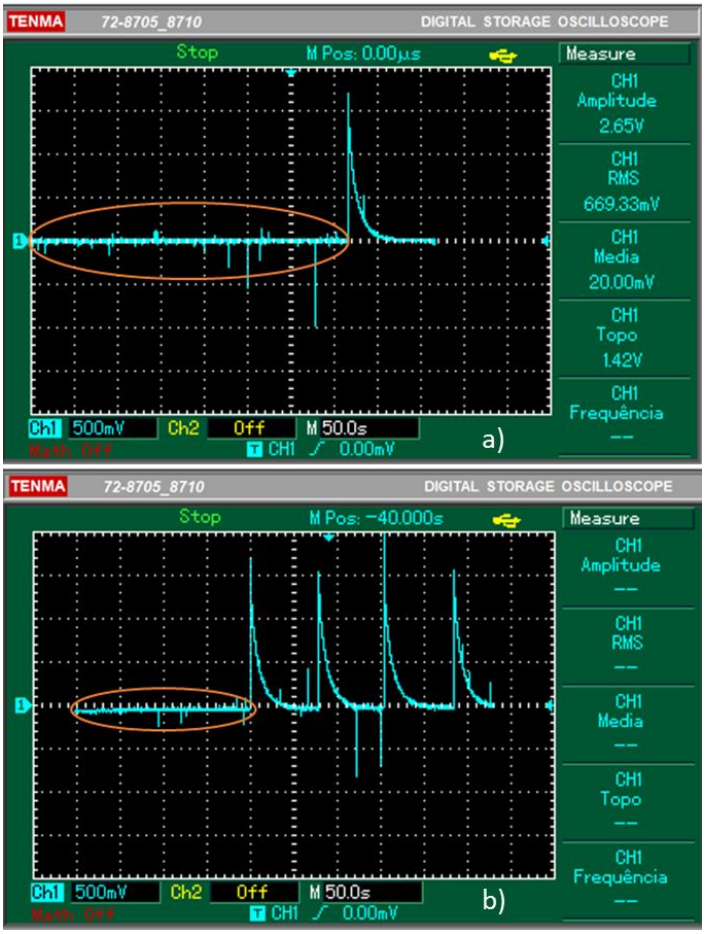


Figura 4.16 – Tempo do primeiro carregamento. a) Transformador cinzento; b) Transformador preto.

É visível uma enorme discrepância entre os dois sensores, em ambos os gráficos, a escala de tempo é de 50 segundos (o valor de cada quadrado no eixo X), tendo isso em consideração é possível saber o tempo total. No gráfico de cima o transformador cinzento demora cerca de 350 segundos até o LED acender pela primeira vez, enquanto o transformador preto necessita de 200 segundos. Após dar o primeiro pulso, a carga existente no condensador caí para valores a rondar os 3.7V voltando a carregar até atingir o suficiente para ativar mais uma vez o LED.

Esse tempo entre picos também foi analisado, na Figura 4.17 são visíveis esses mesmos tempos, o gráfico a) corresponde ao transformador cinzento e o gráfico b) representa o tempo entre picos do LED do transformador preto. Os intervalos de tempo encontram-se devidamente assinalados.

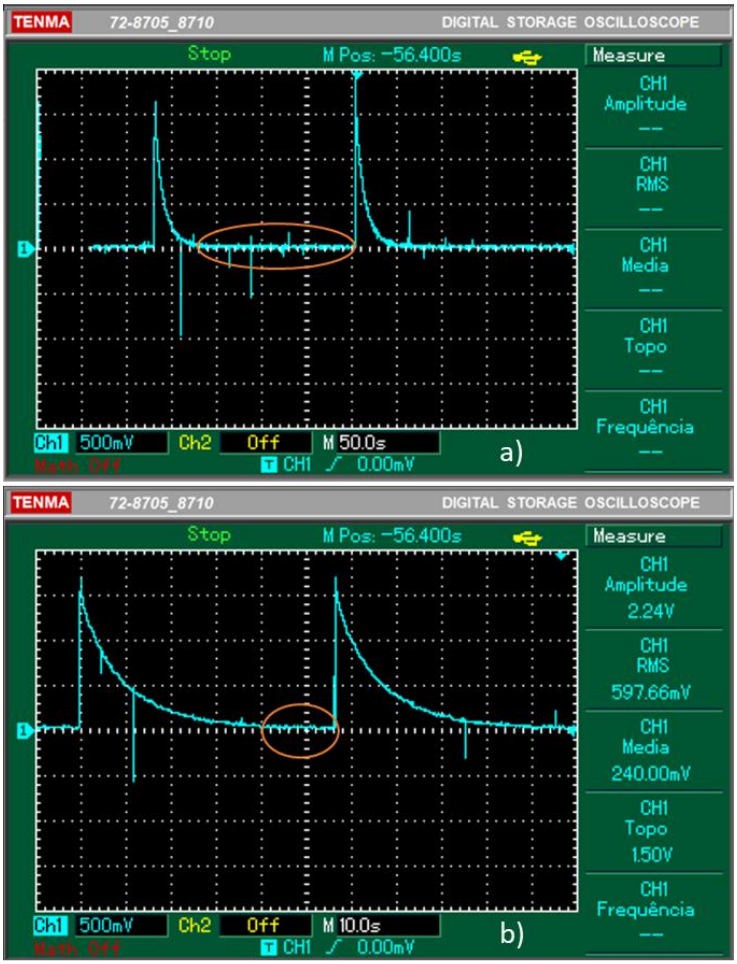


Figura 4.17 – Tempo entre picos de saída. a) Transformador cinzento; b) Transformador preto.

Observou-se uma grande diferença no que respeita ao tempo entre os picos de saída de tensão, utilizando o transformador cinzento. Após o LED acender pela primeira vez, este demora cerca de 160 segundos para que o mesmo volte a acender, ou seja, o condensador só volta atingir a tensão necessária para o LTC3588

disponibilizar à saída passado 160 segundos. No entanto o transformador preto demora muito menos tempo, em cerca de 20 segundos o LED volta a acender o que significa que já existe tensão armazenada pelo integrado.

Avaliou-se também o gráfico sobre o comportamento da tensão para o transformador preto onde a escala do eixo X equivale a 50 segundos de forma a comparar melhor com o gráfico referente ao transformador cinzento, a Figura 4.18 demonstra o resultado obtido.

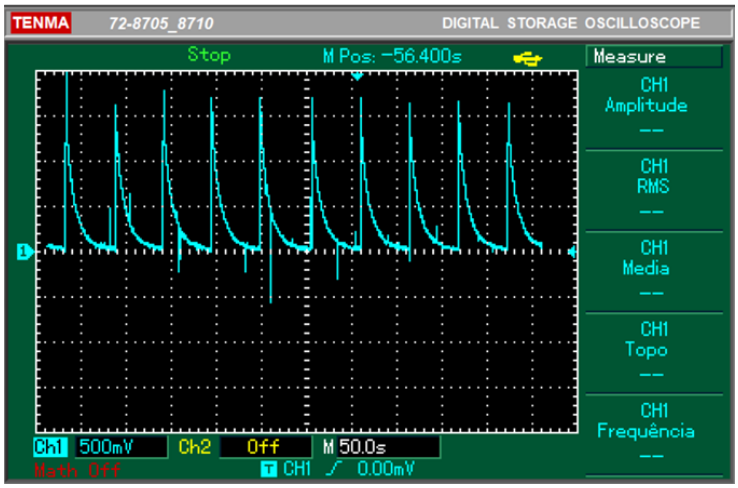


Figura 4.18 – Resultado do transformador de corrente preto com osciloscópio na escala de 50 segundos.

Como foi visto anteriormente, utilizando o transformador de corrente cinzento, o mesmo necessita de 160 segundos para carregar o condensador de forma a permitir um pulso de tensão seguinte, no entanto utilizando o transformador preto no mesmo intervalo de tempo o LED é capaz de acender três vezes, o que é mais uma caraterística a favor do transformador de corrente preto, levando o mesmo a ser escolhido para o estudo sobre a energia reaproveitada no carregamento bem como o impacto, positivo ou negativo, em relação à duração da bateria.

Um tópico relevante para o estudo é o tempo de duração em que a tensão se encontra disponível à saída da placa LTC3588, ou seja, o tempo que o pico de tensão dura ao acender o LED, este teste permite então visualizar e calcular a energia que a saída disponibiliza com a montagem de teste. A região de interesse encontra-se delimitada a vermelho na Figura 4.19, abaixo apresentada.

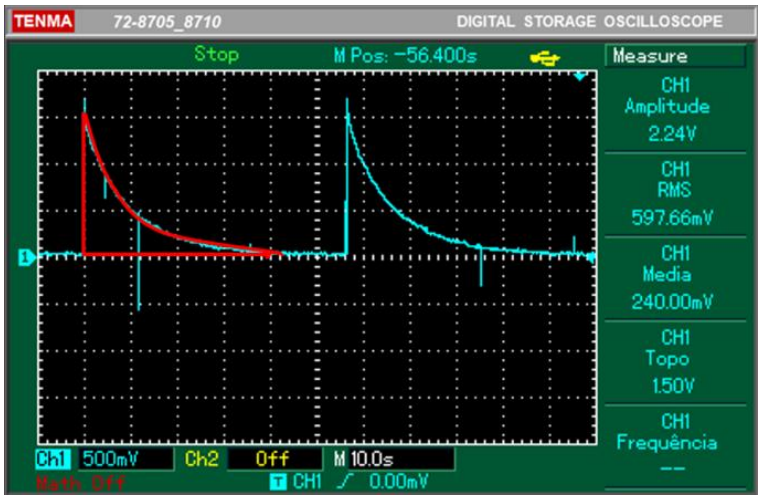


Figura 4.19 – Região onde existe tensão disponível à saída.

No total, a tensão é disponibilizada durante 40 segundos e tendo em conta os 20 segundos necessários para acontecer outro pico, é possível afirmar que existe um pico de minuto em minuto.

4.2.2. Cálculo da energia e do tempo útil da bateria

De seguida, e com a informação proveniente do gráfico anterior, é possível calcular a energia capturada com o transformador sabendo também que o valor da resistência em série com o LED é de 55.4Ω, a tarefa seguinte foi anotar os principais pontos de tensão no gráfico bem como o valor temporal correspondente, de forma que com ajuda da ferramenta Microsoft Excel fosse possível reproduzir a curvatura de tensão, o que é visível na Figura 4.20 onde são visíveis os pares de pontos principais.

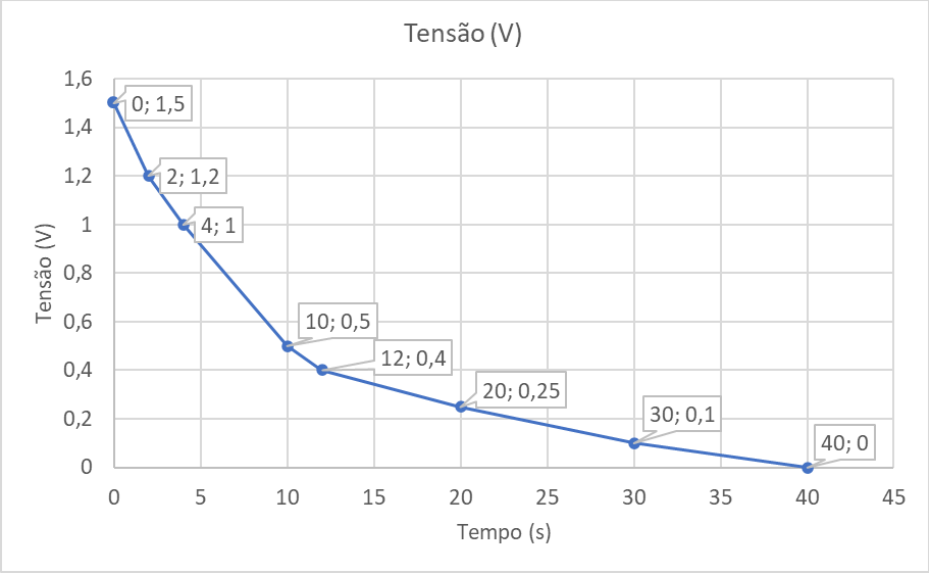


Figura 4.20 – Gráfico do valor de tensão ao longo do tempo.

Através dos pares de pontos selecionados é possível calcular a energia existente através da área do gráfico, por aproximação podemos dividir o gráfico anterior em sete áreas menores, tendo isso em mente de seguida utilizou-se equação 3.

$$Area = \frac{V_0 + V_1}{2 * (s_1 - s_0)} \qquad (Eq. 3)$$

Onde "*V*" significa o valor da tensão em volts e "s" o valor do tempo em segundos dos pares utilizados para o cálculo da área correspondente, ao saber o valor da tensão juntamente com o valor da resistência utilizada é possível calcular o valor da corrente num determinado instante de tempo, o que mais tarde possibilita o cálculo da energia em Joules (1 Joule equivale a 1 watt segundo, Ws) multiplicando o valor da área, anteriormente calculada pelo valor da corrente referente ao mesmo espaço de temporal, a Tabela 4.2 representa os resultados obtidos.

Tabela 4.2 – Resultados obtidos através da análise efetuada.

Tempo (s)	Tensão (V)	Área	Resistência (Ω)	Corrente (A)	Energia J(Ws)
0	1,5	2,7	55,4	0,0271	0,0731
2	1,2	2,2		0,0217	0,0477
4	1	4,5		0,0181	0,0812
10	0,5	0,9		0,0090	0,0081
12	0,4	2,6		0,0072	0,0188
20	0,25	1,75		0,0045	0,0079
30	0,1	0,5		0,0018	0,0009
40	0	------		0,0000	---------
Total		15,15			0,2377

Através dos resultados obtidos o valor da energia total fornecida por um pico de tensão é de 0,2377J. Por fim foi calculada a energia que o transformador de corrente é capaz de produzir num intervalo de uma hora e em cinco horas. A Tabela 4.3 demostra esses mesmos valores.

Tabela 4.3 – Valores de energia ao fim de 1 hora e ao fim de 5 horas.

	Energia J(Ws)
1 hora	14,2608
5 horas	71,3042

O passo seguinte foi simular e calcular a energia que o sistema gasta para efetuar a monitorização, efetuar uma análise completa ao motor e enviar os dados para serem visualizados demora cerca de 20 segundos. Este apresentou um consumo médio de corrente de 120mA sendo este o modo "ativo" do sistema, após esse tempo o sistema entra no modo de *deep sleep*, neste modo o mesmo consome aproximadamente 4mA. Tendo isto em consideração simulou-se e calculou-se a energia consumida pelo sistema caso o mesmo completasse um ciclo de modo "ativo" e *deep sleep* nos seguintes intervalos de tempo:

1 minuto em 1 minuto;

10 minutos em 10 minutos

30 minutos em 30 minutos;

hora em hora;

5 horas em 5 horas.

Os testes foram efetuados em duas situações de alimentação, numa primeira etapa calculou-se para o cenário onde a placa de desenvolvimento seja alimentada pelos 5V (com regulador integrado para os 3.3v) e numa fase mais posterior no caso onde a alimentação do sistema é efetuada aos 3.3V (diretamente sem regulador) tendo como diferença o seu consumo em *deep sleep* sendo ele de 2mA (para os 3.3V).

Para todos os testes o valor de consumo no modo "ativo" é sempre o mesmo visto que tempo não muda, o que difere é o tempo em que o sistema se encontrar em modo *deep sleep*, é também calculado o consumo energético total numa hora, os resultados encontram-se sombreados a azul.

A Tabela 4.4 apresenta os resultados obtidos para o funcionamento num intervalo de tempo de 1 minuto, ou seja, nesta situação permanece 20 segundos no estado "ativo" e os restantes 40 segundo em estado de *deep sleep*.

Tabela 4.4 – Resultados do funcionamento do sistema de minuto em minuto.

Atividade de 1 min em 1 min					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,004	0,02	40	0,80
Trabalhar 1 min	----------	----------	----------	60	12,80
Trabalhar 1h	----------	----------	----------	3600	768

O consumo, no teste de analise e envio dos dados, quando este ocorre de minuto em minuto, estando em modo "ativo" apresenta um consumo de 12J, e no modo *deep sleep*, 0,8J. O que perfaz um total de 12,8J num minuto, convertendo este valor para consumo por hora, o sistema gastará 768J.

O próximo teste consiste apenas no ciclo total de 10 em 10 minutos, ou seja, o sistema encontra-se em modo *deep sleep* durante 580 segundos. Os resultados obtidos, Tabela 4.5, demonstram que ao utilizar este formato, o sistema consume ao fim de 10 minutos 23,6J e ao fim de uma hora consome 141,6J.

Tabela 4.5 – Resultados do funcionamento do sistema de 10 em 10 minutos.

Atividade de 10 min em 10 min					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,004	0,02	580	11,60
Trabalhar 10 min	----------	----------	----------	600	23,60
Trabalhar 1h	----------	----------	----------	3600	141,6

No teste de 30 em 30 minutos, o sistema mantém-se no modo de *deep sleep* durante 1780 segundos, durante esses 30 minutos, o sistema consome perto de 48J, traduzindo isso numa hora corresponde a 95.2J como demonstra a Tabela 4.6.

Tabela 4.6 – Resultados do funcionamento do sistema de 30 em 30 minutos.

Atividade de 30 min em 30 min					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,004	0,02	1780	35,60
Trabalhar 30 min	----------	----------	----------	1800	47,60
Trabalhar 1h	----------	----------	----------	3600	95,2

Os últimos casos referem-se ao funcionamento do sistema uma vez por hora e a cada 5 horas. Em relação ao primeiro caso e após serem feitos os cálculos o sistema consome numa hora 83.6J, na Tabela 4.7 é possível observar esse resultado. Por fim, simulou-se também a operação com maior intervalo, referente ao período de inatividade por parte do sistema, de 5 em 5 horas, o que levou a observar que nesse mesmo tempo, o sistema consume perto de 371,6J dando em média 74.32J, Tabela 4.8.

Tabela 4.7 – Resultados do funcionamento do sistema de hora em hora.

Atividade de 1h em 1h					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,004	0,02	3580	71,60
Trabalhar 1h	----------	----------	----------	3600	83,6

Tabela 4.8 – Resultados do funcionamento do sistema de 5 em 5 horas.

Atividade de 5h em 5h					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,004	0,02	17980	359,60
Trabalhar 5h	----------	----------	----------	18000	371,60
Trabalhar 1h	----------	----------	----------	----------	74,32

Os resultados anteriores são referentes a alimentar o sistema com 5V, no entanto, como foi referido anteriormente foram efetuadas simulações em relação a uma alimentação do sistema utilizando 3.3V, foram utilizados os mesmos cálculos, nas seguintes tabelas, estão apresentados os resultados da energia final consumida pelos diferentes intervalos temporais de funcionamento.

A Tabela 4.9 representa os valores obtidos pelo funcionamento a cada minuto, no qual o sistema consome 8,18J por minuto, que representa em cerca de 491J por hora. Na Tabela 4.10 são visíveis os resultados no processo de funcionamento a cada 10 minutos o que representa 11,75J, ao fim de uma hora traduz-se num consumo de 70,48J no qual se nota uma redução significativa.

A Tabela 4.11 diz respeito ao funcionamento em intervalos de 30 minutos, com um consumo de 19,67J e resulta num consumo de 39,33J ao final de uma hora.

Por fim a Tabela 4.12 representa respetivamente o funcionalmente do sistema de hora em hora, com um consumo de 31,54J. Por fim, a Tabela 4.13 apresenta o funcionamento a cada 5 horas onde o resultado obtido é de 126,59J, em média por hora é de 25,317J.

Tabela 4.9 – Resultado do sistema a funcionar de minuto em minuto.

Atividade de 1 min em 1 min					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	3,3	0,12	0,396	20	7,92
Deep sleep	3,3	0,002	0,0066	40	0,26
Trabalhar 1 min	----------	----------	----------	60	8,18
Trabalhar 1h	----------	----------	----------	3600	491,04

Tabela 4.10 – Resultado do sistema a funcionar de 10 em 10 minutos.

Atividade de 10 min em 10 min					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	3,3	0,12	0,396	20	7,92
Deep sleep	3,3	0,002	0,0066	580	3,83
Trabalhar 10 min	----------	----------	----------	600	11,75
Trabalhar 1h	----------	----------	----------	3600	70,488

Tabela 4.11 – Resultado do sistema a funcionar de 30 em 30 minutos.

Atividade de 30 min em 30 min					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	3,3	0,12	0,396	20	7,92
Deep sleep	3,3	0,002	0,0066	1780	11,75
Trabalhar 30 min	----------	----------	----------	1800	19,67
Trabalhar 1h	----------	----------	----------	3600	39,336

Tabela 4.12 – Resultado do sistema a funcionar de hora em hora.

Atividade de 1h em 1h					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	3,3	0,12	0,396	20	7,92
Deep sleep	3,3	0,002	0,0066	3580	23,63
Trabalhar 1h	----------	----------	----------	3600	31,548

Tabela 4.13 – Resultado do sistema a funcionar de 5 em 5 horas.

Atividade de 5h em 5h					
Tipo:	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	3,3	0,12	0,396	20	7,92
Deep sleep	3,3	0,002	0,0066	17980	118,67
Trabalhar 5h	----------	----------	----------	18000	126,59
Trabalhar 1h	----------	----------	----------	----------	25,3176

Ao efetuar as simulações/testes anteriores, estas permitiram obter os valores de energia que o transformador de corrente é capaz de produzir considerando as condições onde o mesmo foi inserido, bem com a energia utilizada pelo sistema protótipo para realizar a análise e a monitorização em funcionamento intermitente.

Por fim, realizou-se a estimativa do tempo útil da bateria, exemplo de 100mAh (miliampere-hora) caso seja escolhida para alimentar todo o sistema. Seguindo o formato dos testes anteriores, neste caso efetuaram-se cálculos para uma utilização de 5V, mas também com 3.3V, inicialmente calculou-se numa situação sem considerar a energia que o transformador de corrente fornecia e numa segunda vertente contou-se com esse mesmo valor de forma a comparar o tempo útil.

Para o cálculo do tempo útil, utilizou-se os valores consumidos por hora dos diferentes modos de funcionamento, no entanto esses valores de energia estão em watt-segundo (Ws) e foi necessário converter esses valores para watt-hora (Wh). De seguida, com o valor da tensão e da energia em Wh calculou-se o valor da corrente na unidade de ampere-hora (Ah) convertendo depois para a unidade de mAh.

Uma vez obtida a corrente em mAh e considerando o uso de 80% da bateria de 100mAh por motivos de segurança, calculou-se o tempo útil que a mesma poderia funcionar tanto em horas como em dias. Os resultados referentes ao uso tensão de 5V são visíveis na Tabela 4.14 e os dados do uso de 3.3V na Tabela 4.15.

Tabela 4.14 – Resultado sem uso *energy harvesting* e 5V de alimentação.

Resultado com alimentação 5V								
Ciclo de funcionamento	**Energia consumida numa 1h J(Ws)**	**Energia consumida J(Wh)**	**Tensão (V)**	**Corrente consumida (Ah)**	**Corrente consumida (mAh)**	**Duração (H)**	**Duração (H) (80% uso)**	**Duração (Dias)**
1 minuto	768,00	0,2133	5	0,0427	42,6667	2,344	1,8750	0,0781
10 minutos	141,60	0,0393	5	0,0079	7,8667	12,712	10,1695	0,4237
30 minutos	95,2	0,0264	5	0,0053	5,2889	18,908	15,1260	0,6303
1h	83,60	0,0232	5	0,0046	4,6444	21,531	17,2249	0,7177
5h	74,32	0,0206	5	0,0041	4,1289	24,2196	19,3757	0,8073

Tabela 4.15 – Resultado sem uso *energy harvesting* e 3.3V de alimentação.

Resultado com alimentação 3.3V								
Ciclo de funcionamento	**Energia consumida numa 1h J(Ws)**	**Energia consumida J(Wh)**	**Tensão (V)**	**Corrente consumida (Ah)**	**Corrente consumida (mAh)**	**Duração (H)**	**Duração (H) (80% uso)**	**Duração (Dias)**
1 minuto	491,04	0,1364	3,3	0,0413	41,3333	2,4194	1,9355	0,0806
10 minutos	70,49	0,0196	3,3	0,0059	5,9333	16,854	13,483	0,5618
30 minutos	39,336	0,0109	3,3	0,0033	3,3111	30,201	24,161	1,0067
1h	31,55	0,0088	3,3	0,0027	2,6556	37,657	30,126	1,2552
5h	25,32	0,0070	3,3	0,0021	2,1311	46,924	37,539	1,5641

Analisando as duas tabelas anteriores, é possível verificar que ao alimentar com 3.3V o maior tempo útil atingido é de pouco mais que um dia e meio e na menor das hipóteses o sistema não atinge duas horas de operação, o mesmo pode se verificar ao utilizar os 5V sendo que na melhor das hipóteses o sistema dura 19 horas. Para este caso em específico e de forma a validar aos valores da duração útil da bateria, representado nas tabelas anteriores, foram efetuados os cálculos utilizando a equação 4 e equação 5:

$$Tempo = \frac{Capacidade}{Consumo\ Total} * \%\ Bateria\ útil \qquad (Eq.4)$$

$$Consumo\ Total = \frac{(Consumo1 * tempo1 + Consumo2 * tempo2)}{tempo1 + tempo2} \qquad (Eq.5)$$

Na equação do cálculo do "Consumo Total" as variáveis com o terminador "1" correspondem aos valores no modo ativo de funcionamento do sistema, por conseguinte, as variáveis com terminador "2" representam o modo de funcionamento em *deep sleep*.

Em relação ao parâmetro da "%Bateria útil" a escolha de 80% ocorre, novamente, devido a motivos de segurança. Os resultados destes cálculos são visíveis em anexo, o Anexo III representa os valores com uma alimentação de 5V e os resultados com a alimentação de 3.3V são visíveis no Anexo IV.

De seguida simulou-se contabilizando a energia que o transformador fornece, os cálculos foram efetuados apenas a considerar 90% do aproveitamento dessa energia. Neste caso a única mudança comparativamente aos cálculos anteriores é que à energia consumida é subtraída a energia produzida. Na Tabela 4.16 estão representados os dados da alimentação com 5V e na Tabela 4.17 com alimentação de 3.3V.

Tabela 4.16 – Resultado com uso de *energy harvesting* e alimentação 5V.

Uso do *energy harvesting* alimentação 5V										
Ciclo de funcionamento	**Energia consumida numa 1h J(Ws)**	**Energia produzida J(Ws)**	**Energia consumida total J(Ws)**	**Energia consumida J(Wh)**	**Tensão (V)**	**Corrente consumida (Ah)**	**Corrente Consumida (mAh)**	**Duração (H)**	**Duração (H) (80% uso)**	**Duração (Dias)**
1 minuto	768,00	12,8347	755,1653	0,2098	5	0,0420	41,9536	2,3836	1,907	0,0795
10 minutos	141,60	12,8347	128,7653	0,0358	5	0,0072	7,1536	13,9789	11,183	0,4660
30 minutos	95,20	12,8347	82,3653	0,0229	5	0,0046	4,5758	21,8539	17,483	0,7285
1h	83,60	12,8347	70,7653	0,0197	5	0,0039	3,9314	25,4362	20,349	0,8479
5h	74,32	12,8347	61,4853	0,0171	5	0,0034	3,4158	29,2753	23,42	0,9758

Tabela 4.17 – Resultado com uso de energy harvesting e alimentação 3.3V.

Uso do *energy harvesting* alimentação 3.3V										
Ciclo de funcionamento	**Energia consumida numa 1h J(Ws)**	**Energia produzida J(Ws)**	**Energia consumida total J(Ws)**	**Energia consumida J(Wh)**	**Tensão (V)**	**Corrente consumida (Ah)**	**Corrente Consumida (mAh)**	**Duração (H)**	**Duração (H) (80% uso)**	**Duração (Dias)**
1 minuto	491,04	12,8347	478,2053	0,1328	3,3	0,0403	40,2530	2,4843	1,987	0,0828
10 minutos	70,49	12,8347	57,6533	0,0160	3,3	0,0049	4,8530	20,606	16,485	0,6869
30 minutos	39,34	12,8347	26,5013	0,0074	3,3	0,0022	2,2307	44,828	35,862	1,4943
1h	31,55	12,8347	18,7133	0,0052	3,3	0,0016	1,5752	63,484	50,787	2,1161
5h	25,32	12,8347	12,4829	0,0035	3,3	0,0011	1,0507	95,171	76,136	3,1724

O sistema quando alimentado com 5V e com o aproveitamento de energia não se verifica um aumento significativo de uma forma geral comparativamente aos resultados sem o aproveitamento, no entanto com um funcionamento a cada 5h obtemos um aumento de tempo útil de 4 horas, passando de 19 horas para 23 horas.

Alimentando o sistema com 3.3V, os resultados obtidos considerando o uso das técnicas de *energy harvesting*, demonstram um aumento significativo quando comparados com os anteriores resultados onde o maior tempo útil verificado foi no ciclo de funcionamento de 5h em 5h onde se passou de um dia e meio para pouco mais que três dias.

Os resultados obtidos sobre o tempo útil de vida da bateria, são bastante baixos visto que o maior valor simulado é de pouco mais que três dias, um dos motivos para isto ocorrer é que o sistema não utiliza um *hardware* tão atualizado e eficiente no que diz respeito ao *deep sleep*. Para além dos elementos periféricos presentes no módulo terem um foco em custo e não em eficiência, mantendo o consumo em *deep sleep* elevado.

Existem já no mercado outras soluções de *hardware* mais atualizado no qual em *deep sleep* apenas apresenta um consumo de 10μA (microampere) para o mesmo microcontrolador ESP32, tendo em consideração este valor, efetuou-se os testes de consumo e tempo útil da bateria caso fosse utilizado um módulo mais atual.

Os testes/simulações foram efetuados apenas considerando a tensão de alimentação de 5V. Os resultados/simulações do consumo do sistema utilizando o *hardware* com baixo consumo no *deep sleep* são apresentados em modo de tabela. A Tabela 4.18 representa o consumo com o ciclo de análise de minuto em

minuto, a Tabela 4.19 referente a um ciclo a cada 10 minutos, a Tabela 4.20 representa os resultados com intervalos de 30 minutos e por fim a Tabela 4.21 e Tabela 4.22 os ciclos de trabalho de hora em hora e cada 5 horas, respetivamente.

Tabela 4.18 – Resultado do ciclo de minuto em minuto, *deep sleep* 10µA.

Atividade de 1 min em 1 min					
Ciclo de funcionamento	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,00001	0,00005	40	0,00
Trabalhar 1 min	----------	----------	----------	60	12,00
Trabalhar 1h	----------	----------	----------	3600	720,12

Tabela 4.19 – Resultado do ciclo a cada 10 minutos, *deep sleep* 10µA.

Atividade de 10 min em 10 min					
Ciclo de funcionamento	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,00001	0,00005	580	0,03
Trabalhar 10 min	----------	----------	----------	600	12,03
Trabalhar 1h	----------	----------	----------	3600	72,174

Tabela 4.20 – Resultado do ciclo a cada 30 minutos, *deep sleep* 10µA.

Atividade de 30 min em 30 min					
Ciclo de funcionamento	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,00001	0,00005	1780	0,09
Trabalhar 30 min	----------	----------	----------	1800	12,09
Trabalhar 1h	----------	----------	----------	3600	24,178

Tabela 4.21 – Resultado do ciclo de hora a hora, *deep sleep* 10µA.

Atividade de 1h em 1h					
Ciclo de funcionamento	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,00001	0,00005	3580	0,18
Trabalhar 1h	----------	----------	----------	3600	12,179

Tabela 4.22 – Resultado do ciclo de 5 em 5 horas, *deep sleep* 10µA.

Atividade de 5h em 5h					
Ciclo de funcionamento	**Tensão (V)**	**Corrente média (A)**	**Potência (W)**	**Tempo (s)**	**Energia J(Ws)**
Ativo	5	0,12	0,6	20	12,00
Deep sleep	5	0,00001	0,00005	17980	0,90
Trabalhar 5h	----------	----------	----------	18000	12,90
Trabalhar 1h	----------	----------	----------	----------	2,5798

Uma vez calculada a energia consumida por parte do sistema procedeu-se à simulação do tempo útil do mesmo com a utilização de uma bateria de 100mAh, numa primeira instância simulou-se sem a aplicação da técnica de *energy harvesting* nos diversos ciclos de funcionamento, Tabela 4.23 e depois considerando a energia produzida por parte do transformador de corrente, Tabela 4.24.

Tabela 4.23 – Resultado sem recurso a *energy harvesting*.

Sem uso do *energy harvesting* alimentação 5V, *deep sleep* 10µA								
Ciclo de funcionamento	**Energia consumida numa 1h J(Ws)**	**Energia total consumida J(Wh)**	**Tensão (V)**	**Corrente consumida (Ah)**	**Corrente consumida (mAh)**	**Duração (H)**	**Duração (H) (80% uso)**	**Duração (Dias)**
1 minuto	720,12	0,2000	5	0,0400	40,0067	2,4996	1,9997	0,0833
10 minutos	72,17	0,0200	5	0,0040	4,0097	24,9397	19,9518	0,8313
30 minutos	24,18	0,0067	5	0,0013	1,3432	74,4478	59,5583	2,4816
1h	12,18	0,0034	5	0,0007	0,6766	147,7954	118,2363	4,9265
5h	2,58	0,0007	5	0,0001	0,1433	697,7285	558,1828	23,2576

Tabela 4.24 – Resultado com recurso a *energy harvesting*.

Uso do *energy harvesting* alimentação 5V, *deep sleep* 10μA										
Ciclo de funcionamento	**Energia consumida numa 1h J(Ws)**	**Energia produzida J(Ws)**	**Energia consumida total J(Ws)**	**Energia consumida J(Wh)**	**Tensão (V)**	**Corrente consumida (Ah)**	**Corrente consumida (mAh)**	**Duração (H)**	**Duração (H) (80% uso)**	**Duração (Dias)**
1 minuto	720,12	12,8347	707,2853	0,1965	5	0,0392936	39,2936	2,5449	2,036	0,0848
10 minutos	72,17	12,8347	59,3393	0,0165	5	0,003296	3,2966	30,334	24,267	1,0111
30 minutos	24,18	12,8347	11,3433	0,0032	5	0,0006301	0,6302	158,68	126,947	5,2895
1h	12,18	12,8347	-0,6557	-0,0002	5	-0,0000364	-2744,96	∞	∞	∞
5h	2,58	12,8347	-10,2549	-0,0028	5	-0,0005697	-175,53	∞	∞	∞

Através dos resultados, é possível observar diferenças comparativamente ao utilizar o *hardware* com menor consumo, na situação sem o uso da energia fornecida por parte do transformador de corrente, no caso onde ciclo de funcionamento é mais frequente o aumento do tempo útil não é significativo. Porém ao longo dos diferentes ciclos de funcionamento, o tempo útil aumenta de forma considerável atingindo no caso de funcionamento a cada 5 horas um tempo útil de 23 dias onde anteriormente apenas duraria 19 horas.

Os resultados mais otimistas aparecem quando o tempo útil é simulado considerando a energia produzida pelo transformador de corrente, na situação onde o sistema trabalha de minuto em minuto, o tempo de vida útil sofre um aumento de apenas de alguns minutos.

No entanto, ao efetuar uma análise por parte do sistema de 10 em 10 minutos o aumento comparativamente a simulação referente ao uso de técnica de *energy harvesting* com a placa de desenvolvimento utilizado, ao utilizar este *hardware* mais atualizado o tempo útil da bateria sofre um aumento de 13 horas passando de apenas 11 horas de duração para 24 horas, ou seja, um dia. Em relação ao ciclo de funcionamento de 30 minutos em 30 minutos o tempo útil aumenta de apenas 17 horas para 5 dias.

Nos ciclos de funcionamento de hora em hora e de 5 horas em 5 horas, o valor que o sistema consome é menor do que o valor fornecido pelo transformador de corrente, ou seja, com a técnica de *energy harvesting* o sistema é capaz de carregar mais do que necessita para funcionar, tornando o sistema autossuficiente.

Na Tabela 4.24 acima apresentada, existem valores com valor negativo, o que sublinha a afirmação anterior, por exemplo na coluna "Corrente consumida (mAh)", no entanto a corrente consumida não pode ser negativa, ou seja, os valores apresentados indicam que existe maior corrente fornecida à bateria pelo transformador do que consumida em média. Os valores servem também para reforçar o resultado da simulação/teste efetuado.

5. CONCLUSÕES E TRABALHOS FUTUROS

Neste último capítulo são apresentadas conclusões do trabalho realizado sobre dos temas centrais do mesmo, tendo em atenção os resultados obtidos através dos testes efetuados e por fim, são enumeradas algumas tarefas possíveis para trabalho futuro com o intuito de dar seguimento ao projeto desenvolvido de forma aperfeiçoar o sistema e aprimorar os resultados.

5.1. Conclusões

O trabalho presente ao longo deste relatório descreve o desenvolvimento de um sistema embebido em que fosse possível adquirir, monitorizar e analisar/classificar através de uma solução de *tinyML* o modo de funcionamento de um motor AC trifásico utilizando um sensor composto por um acelerómetro e um magnetómetro. Complementado pelo uso de técnicas de *energy harvesting*, para o qual se estudou e simulou o consumo e duração observando o comportamento da vida útil da bateria de forma tornar o sistema autossuficiente *plug-and-play*.

Os objetivos inicialmente propostos foram cumpridos com sucesso, no caso do modelo desenvolvido os resultados de previsão obtidos ao receber 100 novos conjuntos de valores demonstrou-se bastante positiva, de 97.26 %. É importante referir que o modelo é do tipo *supervised learning* o que implica que o mesmo só apresenta eficácia na configuração do motor definida, ou seja, caso o motor seja alterado ou a aplicação, a performance do mesmo iria baixar significativamente.

Em relação ao tempo útil da bateria e ao estudo efetuado, a aplicação de *energy harvesting* apresentou-se como uma solução aplicável, ao implementar com o *hardware* disponível para a aplicação da rede neuronal observou-se que o tempo de vida sofria um aumento, mesmo que marginal, mas ficando um pouco aquém das espectativas em relação ao tempo total útil.

No entanto, simulou-se para o uso de um *hardware* eficiente que permite um reduzido consumo em estado de *deep sleep*, no qual se mostrou como uma solução viável e que em certos casos é possível obter um sistema autónomo e em alguns casos com um tempo de vida útil infinito. Porém é de elevada importância realçar que a placa LTC3588 trabalha com valores entre 2.7V a 20V que se mostram bastante superiores ao fornecido pelo transformador de corrente de apenas 400mV o que significa que a placa não se encontra a funcionar a 100% no melhor regime de eficiência.

Ao longo do desenvolvimento deste projeto os problemas e as dificuldades foram aparecendo, nos dois temas abordados, mas foi na criação do modelo e na sua aplicação onde os obstáculos se fizeram sentir mais devido a ser um tema no qual não estava habituado. Uma limitação na parte do *energy harvesting*, como já foi referido anteriormente, é o valor necessário de tensão de entrada que o circuito integrado necessita para funcionar normalmente que não era atingido.

Através desta prova de conceito verificou-se a fiabilidade do uso do modelo preditivo num ambiente laboratorial o que prova que o seu uso no ambiente industrial é possível de acontecer e que é viável criar um sistema que se alimenta sozinho criando assim o sistema de *plug-and-play*, que era um dos objetivos de estudo na presente dissertação.

5.2. Trabalhos futuros

Este projeto, como já foi mencionado, foi desenvolvido num formato de prova de conceito, e como trabalho futuro será dar continuidade ao mesmo criando um sistema final e testado em ambiente industrial, para isso a nível de *hardware* será necessário criar uma PCB (*Prints Circuit Board*) utilizando componentes com um baixo consumo em estado de *deep sleep* de forma a atingir um tempo de vida útil alto.

Em relação ao modelo preditivo propõe-se como melhoria implementar técnicas de *unsupervised learning* de maneira a combater a fragilidade da rede atual que necessita que o sistema seja acoplado de uma maneira sistemática.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] "INDUSTRIAL REVOLUTION 4.0 - Concept N Content." [Online]. Available: https://www.conceptncontent.com/industrial-revolution-4-0/. [Accessed: 18-Mar-2021].

[2] E. G. Popkova, Y. V Ragulina, and A. V Bogoviz, "Fundamental differences of transition to industry 4.0 from previous industrial revolutions," in *Studies in Systems, Decision and Control*, vol. 169, 2019, pp. 21–29.

[3] A. Cachada *et al.*, "Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, vol. 2018-September, pp. 139–146, Oct. 2018, doi: 10.1109/ETFA.2018.8502489.

[4] G. S. Sampaio, A. R. de A. V. Filho, L. S. da Silva, and L. A. da Silva, "Prediction of motor failure time using an artificial neural network," *Sensors (Switzerland)*, vol. 19, no. 19, p. 4342, Oct. 2019, doi: 10.3390/s19194342.

[5] P. Poo´r, D. Zˇeni´sˇek, and J. Basl, "Historical overview of maintenance management strategies: Development from breakdown maintenance to predictive maintenance in accordance with four industrial revolutions," in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2019, no. July, pp. 495–504.

[6] A. Bousdekis, D. Apostolou, and G. Mentzas, "Predictive Maintenance in the 4th Industrial Revolution: Benefits, Business Opportunities, and Managerial Implications," *IEEE Eng. Manag. Rev.*, vol. 48, no. 1, pp. 57–62, Jan. 2020, doi: 10.1109/EMR.2019.2958037.

[7] C. Covaci and A. Gontean, "Piezoelectric Energy Harvesting Solutions: A Review," *Sensors 2020, Vol. 20, Page 3512*, vol. 20, no. 12, p. 3512, Jun. 2020, doi: 10.3390/S20123512.

[8] B. Mahesh, "Machine Learning Algorithms-A Review," *Int. J. Sci. Res.*, 2018, doi: 10.21275/ART20203995.

[9] R. Justo-Silva, A. Ferreira, and G. Flintsch, "Review on Machine Learning Techniques for Developing Pavement Performance Prediction Models," *Sustain. 2021, Vol. 13, Page 5248*, vol. 13, no. 9, p. 5248, May 2021, doi: 10.3390/SU13095248.

[10] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, pp. 1–16, Dec. 2019, doi: 10.1186/S12911-019-1004-8/FIGURES/12.

[11] G. Cho, J. Yim, Y. Choi, J. Ko, and S. H. Lee, "Review of machine learning algorithms for diagnosing mental illness," *Psychiatry Investig.*, vol. 16, no. 4, pp. 262–269, Apr. 2019, doi: 10.30773/pi.2018.12.21.2.

[12] S. Ray, "A Quick Review of Machine Learning Algorithms," in *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, 2019, pp. 35–39, doi: 10.1109/COMITCon.2019.8862451.

[13] "A Deep Dive into Decision Trees. A decision tree is a technique of… | by Soumo Chatterjee | Analytics Vidhya | Medium." [Online]. Available: https://medium.com/analytics-vidhya/a-deep-dive-to-decision-trees-6575e016d656.

[14] "Máquina de vetores de suporte (SVM) explicada." [Online]. Available: https://ichi.pro/pt/maquina-de-vetores-de-suporte-svm-explicada-97743104690915.

[15] M. Usama *et al.*, "Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges," *IEEE Access*, vol. 7, pp. 65579–65615, 2019, doi: 10.1109/ACCESS.2019.2916648.

[16] T. Kurita, "Principal Component Analysis (PCA)," *Comput. Vis.*, pp. 1–4, 2020, doi: 10.1007/978-3-030-03243-2_649-1.

[17] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electron. 2020, Vol. 9, Page 1295*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: 10.3390/ELECTRONICS9081295.

[18] "Para começar com agrupamento K-Means." [Online]. Available: https://ichi.pro/pt/para-comecar-com-agrupamento-k-means-31262593745935.

[19] Q. Wang and Z. Zhan, "Reinforcement learning model, algorithms and its application," *Proc. 2011 Int. Conf. Mechatron. Sci. Electr. Eng. Comput. MEC 2011*, pp. 1143–1146, 2011, doi:

10.1109/MEC.2011.6025669.
[20] "Introduction to Reinforcement Learning for Beginners." [Online]. Available: https://www.analyticsvidhya.com/blog/2021/02/introduction-to-reinforcement-learning-for-beginners/.
[21] A. D. Dongare, R. R. Kharde, and A. D. Kachare, "Introduction to Artificial Neural Network," *Certif. Int. J. Eng. Innov. Technol.*, vol. 9001, no. 1, pp. 2277–3754, 2008.
[22] "Applied Deep Learning - Part 1: Artificial Neural Networks | by Arden Dertat | Towards Data Science." [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6.
[23] S. Leijnen and F. van Veen, "The Neural Network Zoo," *Proceedings*, vol. 47, no. 1, p. 9, May 2020, doi: 10.3390/proceedings2020047009.
[24] "What are Neural Networks? | IBM." [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks#toc-types-of-n-YgdI1-Kt.
[25] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-January, pp. 1–6, Mar. 2018, doi: 10.1109/ICENGTECHNOL.2017.8308186.
[26] "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science." [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.
[27] D. L. Dutta and S. Bharali, "TinyML Meets IoT: A Comprehensive Survey," *Internet of Things (Netherlands)*, vol. 16, p. 100461, Dec. 2021, doi: 10.1016/j.iot.2021.100461.
[28] R. Sanchez-Iborra and A. F. Skarmeta, "TinyML-Enabled Frugal Smart Objects: Challenges and Opportunities," *IEEE Circuits Syst. Mag.*, vol. 20, no. 3, pp. 4–18, 2020, doi: 10.1109/MCAS.2020.3005467.
[29] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *J. King Saud Univ. - Comput. Inf. Sci.*, Nov. 2021, doi: 10.1016/j.jksuci.2021.11.019.
[30] "TinyML: Machine Learning for Embedded System — Part I | by Leonardo Cavagnis | Medium." [Online]. Available: https://leonardocavagnis.medium.com/tinyml-machine-learning-for-embedded-system-part-i-92a34529e899.
[31] M. Shafique, T. Theocharides, V. J. Reddy, and B. Murmann, "TinyML: Current Progress, Research Challenges, and Future Roadmap," *Proc. - Des. Autom. Conf.*, vol. 2021-Decem, pp. 1303–1306, Dec. 2021, doi: 10.1109/DAC18074.2021.9586232.
[32] R. Jigyasu, A. Sharma, L. Mathew, and S. Chatterji, "A Review of Condition Monitoring and Fault Diagnosis Methods for Induction Motor," in *Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018*, 2019, pp. 1713–1721, doi: 10.1109/ICCONS.2018.8662833.
[33] O. E. Hassan, M. Amer, A. K. Abdelsalam, and B. W. Williams, "Induction motor broken rotor bar fault detection techniques based on fault signature analysis – a review," *IET Electr. Power Appl.*, vol. 12, no. 7, pp. 895–907, Aug. 2018, doi: 10.1049/iet-epa.2018.0054.
[34] Y. Liu and A. M. Bazzi, "A review and comparison of fault detection and diagnosis methods for squirrel-cage induction motors: State of the art," *ISA Trans.*, vol. 70, pp. 400–409, Sep. 2017, doi: 10.1016/j.isatra.2017.06.001.
[35] K. N. Gyftakis, P. A. Panagiotou, and S. Bin Lee, "The Role of the Mechanical Speed Frequency on the Induction Motor Fault Detection via the Stray Flux," in *Proceedings of the 2019 IEEE 12th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, SDEMPED 2019*, 2019, pp. 201–207, doi: 10.1109/DEMPED.2019.8864863.
[36] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7067–7075, Nov. 2016, doi: 10.1109/TIE.2016.2582729.
[37] A. Harb, "Energy harvesting: State-of-the-art," *Renew. Energy*, vol. 36, no. 10, pp. 2641–2654, Oct. 2011, doi: 10.1016/j.renene.2010.06.014.
[38] N. Garg and R. Garg, "Energy harvesting in IoT devices: A survey," in *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017*, 2018, pp. 127–131, doi: 10.1109/ISS1.2017.8389371.
[39] P. Choudhary, L. Bhargava, V. Singh, M. Choudhary, and A. kumar Suhag, "A survey - Energy

harvesting sources and techniques for internet of things devices," *Mater. Today Proc.*, vol. 30, pp. 52–56, 2020, doi: 10.1016/J.MATPR.2020.04.115.

[40] I. Sil, S. Mukherjee, and K. Biswas, "A review of energy harvesting technology and its potential applications," *Environ. EARTH Sci. Res. J.*, vol. 4, no. 2, pp. 33–38, Jun. 2017, doi: 10.18280/EESRJ.040202.

[41] V. Annapureddy *et al.*, "Magnetic energy harvesting with magnetoelectrics: An emerging technology for self-powered autonomous systems," *Sustain. Energy Fuels*, vol. 1, no. 10, pp. 2039–2052, Nov. 2017, doi: 10.1039/c7se00403f.

[42] F. Yang, L. Du, H. Yu, and P. Huang, "Magnetic and Electric Energy Harvesting Technologies in Power Grids: A Review," *Sensors*, vol. 20, no. 5, p. 1496, Mar. 2020, doi: 10.3390/s20051496.

[43] P. Hawliczek, D. V. Tien, and R. Gono, "Creating and measuring unipolar and bipolar ultra-low voltage step-up converters," *Proc. 2017 18th Int. Sci. Conf. Electr. Power Eng. EPE 2017*, Jun. 2017, doi: 10.1109/EPE.2017.7967308.

[44] "GitHub - metanav/Handwriting_Recognition_IMU: Handwriting recognition using Sparkfun Artemis ATP." [Online]. Available: https://github.com/metanav/Handwriting_Recognition_IMU.

[45] D. Dinculeană and X. Cheng, "Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices," *Appl. Sci. 2019, Vol. 9, Page 848*, vol. 9, no. 5, p. 848, Feb. 2019, doi: 10.3390/APP9050848.

[46] G. Sasikala *et al.*, "IoT real time data acquisition using MQTT protocol," *J. Phys. Conf. Ser.*, vol. 853, no. 1, p. 012003, May 2017, doi: 10.1088/1742-6596/853/1/012003.

[47] I. Hedi, I. Špeh, and A. Šarabok, "IoT network protocols comparison for the purpose of IoT constrained networks," *2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc.*, pp. 501–505, Jul. 2017, doi: 10.23919/MIPRO.2017.7973477.

[48] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," *2017 IEEE Int. Symp. Syst. Eng. ISSE 2017 - Proc.*, Oct. 2017, doi: 10.1109/SYSENG.2017.8088251.

[49] S. Hamdani and H. Sbeyti, "A comparative study of COAP and MQTT communication protocols," *7th Int. Symp. Digit. Forensics Secur. ISDFS 2019*, Jun. 2019, doi: 10.1109/ISDFS.2019.8757486.

[50] M. Lekić and G. Gardašević, "IoT sensor integration to Node-RED platform," *2018 17th Int. Symp. INFOTEH-JAHORINA, INFOTEH 2018 - Proc.*, vol. 2018-January, pp. 1–5, Apr. 2018, doi: 10.1109/INFOTEH.2018.8345544.

[51] I.-V. Nițulescu and A. Korodi, "Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions," *IoT 2020, Vol. 1, Pages 76-91*, vol. 1, no. 1, pp. 76–91, Aug. 2020, doi: 10.3390/IOT1010005.

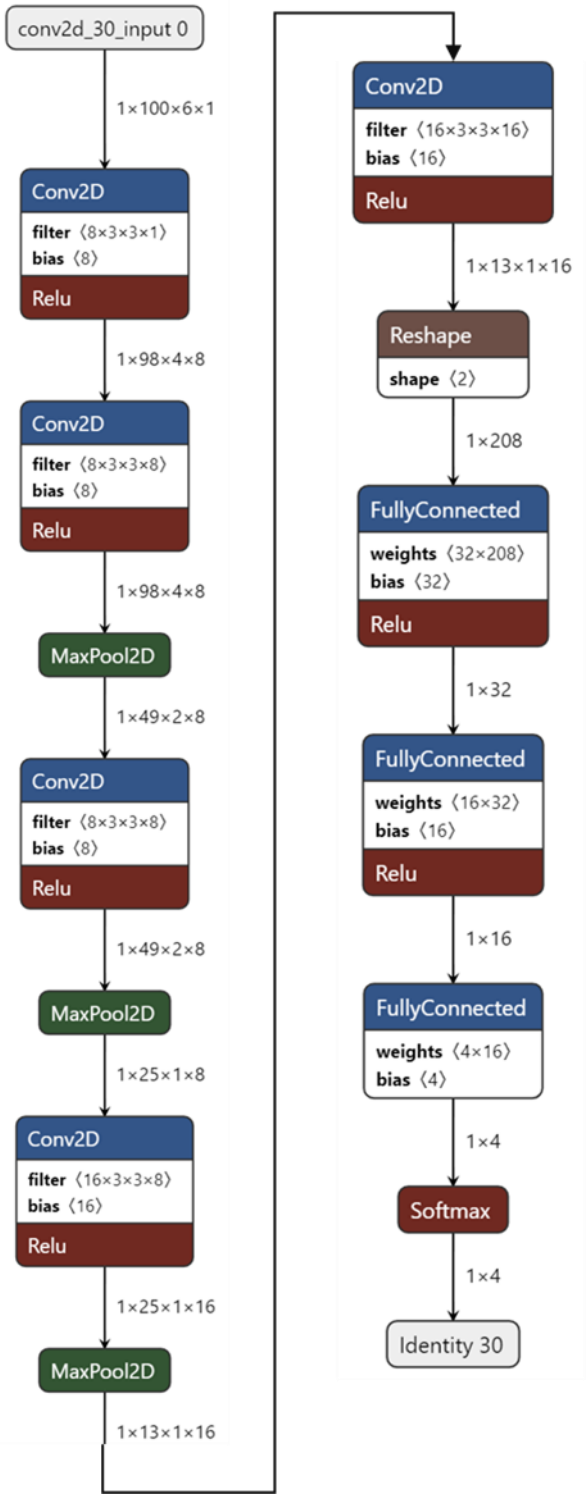
[52] M. A. Kausar, "Suitability Of Influxdb Database For IoT Applications," *Artic. Int. J. Innov. Technol. Explor. Eng.*, no. 10, pp. 2278–3075, 2019, doi: 10.35940/ijitee.J9225.0881019.

[53] S. Di Martino, L. Fiadone, A. Peron, V. N. Vitale, and A. Riccabone, "Industrial Internet of Things: Persistence for Time Series with NoSQL Databases," *Proc. - 2019 IEEE 28th Int. Conf. Enabling Technol. Infrastruct. Collab. Enterp. WETICE 2019*, pp. 340–345, Jun. 2019, doi: 10.1109/WETICE.2019.00076.

[54] A. Raza, A. Ul, and N. Shahani, "A Survey on Internet of Things and Databases Annals of Emerging Technologies in Computing (AETiC) View project Technical SCIENCE ⚙ View project," *J. Inf. Commun. Technol.*, vol. 14, no. 1.

ANEXOS

ANEXO I ESTRUTURA DA REDE NEURONAL UTILIZADA



ANEXO II RESULTADO DAS CLASSIFICAÇÕES

Modo	Distribuição carga	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6	Teste 7	Teste 8	Teste 9	Teste 10
		[Normal],[Anomalia],[Parado],[Anomalia Obstáculo]									
Anomalia	1 – 0	[0],[19],[0],[0]	[0],[19],[0],[0]	[8],[11],[0],[0]	[3],[16],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[1],[18],[0],[0]	[0],[19],[0],[0]
	2 – 0	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]
	1 – 2	[0],[19],[0],[0]	[0],[19],[0],[0]	[1],[18],[0],[0]	[0],[19],[0],[0]	[2],[17],[0],[0]	[1],[18],[0],[0]	[2],[17],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[4],[15],[0],[0]
	1 – 3	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]
	2 – 3	[1],[18],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[2],[17],[0],[0]	[1],[18],[0],[0]	[0],[19],[0],[0]	[1],[18],[0],[0]	[1],[18],[0],[0]	[1],[18],[0],[0]	[0],[19],[0],[0]
	2 – 4	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]	[0],[19],[0],[0]
Normal	1 – 1	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[18],[0],[0],[1]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]
	2 – 2	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]
	3 – 3	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]
	0 – 0	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]	[19],[0],[0],[0]
Parado	Parado	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]	[0],[0],[19],[0]
Anomalia Obstaculo	1 – 1	[2],[0],[0],[17]	[1],[0],[0],[18]	[0],[0],[0],[19]	[5],[0],[0],[14]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[4],[0],[0],[15]	[1],[0],[0],[18]	[1],[0],[0],[18]
	2 – 2	[1],[0],[0],[18]	[0],[0],[0],[19]	[0],[0],[0],[19]	[4],[0],[0],[15]	[1],[0],[0],[18]	[0],[0],[0],[19]	[1],[0],[0],[18]	[0],[0],[0],[19]	[4],[0],[0],[15]	[0],[0],[0],[19]
	3 – 3	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[1],[0],[0],[18]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]
	0 – 0	[1],[0],[0],[18]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[0],[0],[0],[19]	[2],[0],[0],[17]	[1],[0],[0],[18]	[0],[0],[0],[19]

ANEXO III CÁLCULO TEMPO DE VIDA ÚTIL COM TENSÃO 5V

Atividade de 1 min em 1 min							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	42,6667	1,8750	0,078125
Deep sleep	0,004	4	40	160			
Trabalhar 1 min	----------	----------	60	2560			

Atividade de 10 min em 10 min							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	7,8667	10,1695	0,42372
Deep sleep	0,004	4	580	2320			
Trabalhar 10 min	----------	----------	600	4720			

Atividade de 30 min em 30 min							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	5,2889	15,1261	0,63025
Deep sleep	0,004	4	1780	7120			
Trabalhar 30 min	----------	----------	1800	9520			

Atividade de 1h em 1h							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	4,6444	17,2249	0,7177
Deep sleep	0,004	4	3580	14320			
Trabalhar 1h	----------	----------	3600	16720			

Atividade de 5h em 5h							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	4,1289	19,3757	0,8073
Deep sleep	0,004	4	17980	71920			
Trabalhar 5h	----------	----------	18000	74320			

ANEXO IV CÁLCULO TEMPO DE VIDA ÚTIL COM TENSÃO 3.3V

Atividade de 1 min em 1 min							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	41,3333	1,9355	0,08064
Deep sleep	0,002	2	40	80			
Trabalhar 1 min	----------	----------	60	2480			

Atividade de 10 min em 10 min							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	5,9333	13,4831	0,56179
Deep sleep	0,002	2	580	1160			
Trabalhar 10 min	----------	----------	600	3560			

Atividade de 30 min em 30 min							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	3,3111	24,1611	1,0067
Deep sleep	0,002	2	1780	3560			
Trabalhar 30 min	----------	----------	1800	5960			

Atividade de 1h em 1h							
Tipo:	**Corrente média (A)**	**Corrente media (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	2,6556	30,1255	1,2552
Deep sleep	0,002	2	3580	7160			
Trabalhar 1 Hora	----------	----------	3600	9560			

Atividade de 5h em 5h							
Tipo:	**Corrente média (A)**	**Corrente média (mA)**	**Tempo (s)**	**Consumo médio**	**Consumo Total**	**Duração (horas)**	**Duração (dias)**
Ativo	0,12	120	20	2400	2,1311	37,5391	1,5641
Deep Sleep	0,002	2	17980	35960			
Trabalhar 5 Horas	----------	----------	18000	38360			