

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

BRUNA MARTINI

**ESTUDO DE UM SISTEMA PARA O
RECONHECIMENTO DE CARGAS ELÉTRICAS
LIGADAS À REDE**

FLORIANÓPOLIS, 2020

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

BRUNA MARTINI

**ESTUDO DE UM SISTEMA PARA O
RECONHECIMENTO DE CARGAS ELÉTRICAS
LIGADAS À REDE**

Trabalho de conclusão de curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de engenheiro eletrônico

Orientador:
Prof. Daniel Lohmann

FLORIANÓPOLIS, 2020

Ficha de identificação da obra elaborada pelo autor.

Martini, Bruna

ESTUDO DE UM SISTEMA PARA O RECONHECIMENTO DE CARGAS ELÉTRICAS LIGADAS À REDE / Bruna Martini ; orientação de Daniel Lohmann. - Florianópolis, SC, 2020.

66 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Eletrônica. Departamento Acadêmico de Eletrônica.

Inclui Referências.

1. Identificação de cargas. 2. Inteligência artificial.
3. Rede elétrica. 4. Assinatura de cargas. I. Lohmann, Daniel. II. Instituto Federal de Santa Catarina. Departamento Acadêmico de Eletrônica. III. Título.

ESTUDO DE UM SISTEMA PARA O RECONHECIMENTO DE CARGAS ELÉTRICAS LIGADAS À REDE

BRUNA MARTINI

Este trabalho foi julgado adequado para obtenção do Título de Engenheiro Eletrônico em outubro de 2020 e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Eletrônica do Instituto Federal de Educação Ciência, e Tecnologia de Santa Catarina.

Florianópolis, 15 de Outubro de 2020.

Banca Examinadora:

Daniel Lohmann, Me.

Reginaldo Steinbach, Me.

Renan Augusto Starke, Dr.

AGRADECIMENTOS

A Deus por me proporcionar força para não desistir durante essa jornada.

Ao meu orientador, Daniel Lohmann, por aceitar me conduzir nesta pesquisa e por toda paciência e dedicação durante elaboração do meu trabalho de conclusão de curso.

A professora Cláudia, que me conduziu na escrita durante o trabalho de conclusão de curso.

A todos os professores, por todo apoio e conselho durante o curso, e por me proporcionarem um ambiente propício para o desenvolvimento do meu trabalho.

Ao Lucas, meu noivo, por sempre me ajudar, compreender e apoiar.

Aos meus pais por estarem sempre presentes e por todo apoio demonstrado durante esse período tão importante da minha vida acadêmica.

A minha irmã, Thaine, por toda ajuda apresentada durante o curso e o TCC.

As alunas Giovanna, Maria Claudia e Thaine pela ajuda na elaboração do tema desenvolvido neste trabalho.

A todas as pessoas que direta ou indiretamente contribuíram para a realização da minha pesquisa.

RESUMO

A identificação de cargas lineares e não lineares presentes no sistema de energia elétrica tem a finalidade de promover um maior controle do consumo dos aparelhos, identificar anomalias e gerar economias a partir do conhecimento destes dados. O trabalho de conclusão de curso que se segue trata-se do estudo de assinaturas de cargas residenciais, e a definição de um método, eficiente, para o reconhecimento das assinaturas de consumo elétrico. Para isso, foi utilizado o conhecimento adquirido durante o curso de graduação de engenharia eletrônica do Instituto Federal de Santa Catarina para desenvolver as chamadas assinaturas de cargas, que, em outras palavras, são um conjunto de dados de um determinado aparelho, geralmente compostos por parâmetros como: tensão, corrente, potências ativas e as harmônicas do sinal de corrente. Com as assinaturas de cargas já conhecidas pode-se então fazer a análise dos parâmetros presentes na rede elétrica, e assim, através de processamento digital de sinais e inteligência artificial é possível compará-los com as assinaturas já conhecidas e distinguir o aparelho que está ligado à rede.

Palavras-chaves: Identificação de cargas. Inteligência artificial. Rede elétrica. Assinatura de cargas.

ABSTRACT

The identification of linear and non-linear loads present in the electric power system has the purpose of promoting greater control of the consumption of the devices, identifying anomalies and generating savings from the knowledge of these data. This undergraduate thesis is the study of signatures for residential loads and the definition of an efficient method for the recognition of signatures for electrical consumption. For this, it was used the knowledge acquired during the electronic engineering undergraduate course at the Federal Institute of Santa Catarina to develop the so-called load signatures, which in other words are a set of data for a given device, usually composed of parameters such as: voltage, current, active powers and the harmonics of the current signal. With the already known load signatures, it is then possible to analyze the parameters present in the electrical power system, therefore, through digital signal processing and artificial intelligence it is possible to compare them with the already known signatures and distinguish the device that is connected the grid.

Key-Words: Load identification. Artificial intelligence. Electrical power system. Load signature.

LISTA DE ILUSTRAÇÕES

Figura 1 – Consumo de energia agregado a vários dispositivos distintos	20
Figura 2 – Sensor ACS712	21
Figura 3 – Circuito para o sensor de corrente	22
Figura 4 – Circuito divisor de tensão	23
Figura 5 – EVM430-F6736	25
Figura 6 – Diagrama de blocos MSP430F673X	25
Figura 7 – Imagem ilustrativa de uma árvore de decisão	29
Figura 8 – Imagem ilustrativa de uma floresta aleatória	32
Figura 9 – Ilustração do produto instalado	35
Figura 10 – Primeiros testes na captura do sinal na bancada	36
Figura 11 – Fluxograma do <i>software</i> total	37
Figura 12 – Gráficos da corrente RMS e potência de uma cafeteira em funcionamento .	41
Figura 13 – Gráficos da corrente RMS e potência de uma lâmpada	42
Figura 14 – Gráfico da corrente RMS de um micro-ondas em funcionamento	43
Figura 15 – Quantidade de linhas e colunas	46
Figura 16 – Transformação de valores em variáveis	46
Figura 17 – Importação da biblioteca e conversão para matrizes	47
Figura 18 – Divisão em parâmetros de teste e de treinamento	48
Figura 19 – Precisão obtida no primeiro teste do código <i>Random Forest</i>	48
Figura 20 – Precisão obtida no segundo teste do código <i>Random Forest</i>	49
Figura 21 – Precisão obtida no terceiro teste do código <i>Random Forest</i>	49
Figura 22 – Precisão obtida no quarto teste do código <i>Random Forest</i>	51

LISTA DE TABELAS

Tabela 1 – Tabela ilustrativa	30
Tabela 2 – Dados médios	45

LISTA DE ABREVIATURAS E SIGLAS

EPE	Empresa de Pesquisa Energética
LED	<i>Light Emitting Diode</i>
IFSC	Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina
VDC	<i>Voltage Direct Current</i> - Tensão Contínua
VAC	<i>Voltage Alternating Current</i> - Tensão Alternada
CELESC	Centrais Elétricas de Santa Catarina
WI-FI	<i>Tecnologia Wireless da marca Wi-Fi Alliance</i>
ACEEE	<i>American Council for an Energy-Efficient Economy</i>
NILM	<i>Nonintrusive Load Monitoring</i>
ILM	<i>Intrusive Load Monitoring</i>
USART	<i>Universal Synchronous Asynchronous Receiver Transmitter</i>
CSV	<i>Comma Separated Values</i>

LISTA DE SÍMBOLOS

V	Volts - Unidade de potencial elétrico
A	Ampere - Unidade de Corrente Elétrica
Ω	Ohms - Unidade de resistência elétrica
W	Watt's - Unidade de potência elétrica
Hz	Hertz - Unidade de Frequência (Ciclos por Segundo)
Var	Volt-ampere reativo - Unidade de potência reativa
F	Farad - Unidade de Capacitância

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	16
1.2	Definições do problema	17
1.3	Objetivo Geral	17
1.4	Objetivos Específicos	17
2	REVISÃO BIBLIOGRÁFICA	18
2.1	Energia elétrica	18
2.2	Eficiência Energética	18
2.3	Sistemas de Monitoramento	19
2.4	Identificação de carga não intrusiva	19
2.5	Medição de Corrente	20
2.5.1	Efeito Hall	20
2.5.2	Transformador de Corrente	21
2.5.3	Resistência <i>Shunt</i>	22
2.6	Divisor de Tensão	23
2.7	Aquisição da Potência Ativa e Reativa	24
2.8	Taxa de Distorção Harmônica	24
2.9	EVM430-F6736	24
2.10	Processamento de Sinais	26
2.10.1	Processamento analógico de sinal	26
2.10.2	Processamento digital de sinal	26
2.11	Assinatura de Cargas	26
2.12	Banco de Dados	27
2.13	Aprendizagem de Máquina	28
2.13.1	Árvore de Decisão	28
2.13.2	Naive Bayes	30
2.13.3	<i>Random Forest</i>	31
2.14	Linguagens de Programação	33
2.14.1	Python	33
3	METODOLOGIA	34
4	DESCRIÇÃO DO SISTEMA	35
5	RECONHECIMENTO E IDENTIFICAÇÃO DO ELETRODOMÉSTICO	39

6	CONSIDERAÇÕES FINAIS	52
	REFERÊNCIAS	54
	APÊNDICE A – CÓDIGO UTILIZADO PARA SALVAR OS DA- DOS EM ARQUIVO CSV	57
	APÊNDICE B – CÓDIGO UTILIZADO PARA O CÁLCULO DO GANHO DE DECISÃO	59
	APÊNDICE C – SEGUNDO TESTE DA PARA O MÉTODO DA ÁRVORE DE DECISÃO	61
	APÊNDICE D – CÓDIGO UTILIZADO PARA O MÉTODO RAN- DOM FOREST	63
	APÊNDICE A – CÓDIGO UTILIZADO PARA EXTRAIR OS DA- DOS DO BANCO	65

1 INTRODUÇÃO

O consumo de energia elétrica vem aumentando cada vez mais; segundo a Resenha Mensal do Consumo de Energia (2019), divulgada pela EPE (Empresa de Pesquisa Energética), o aumento no primeiro bimestre de 2019 foi de 4,7% comparado ao mesmo período do ano anterior. Essa expansão acentuada vem agregada de várias preocupações, como a possibilidade de esgotamento dos recursos utilizados para a sua produção, a questão econômica, aliada à eficiência energética e os impactos ambientais (FILHO; AZEVEDO, 2013).

A possibilidade do esgotamento de recursos naturais é real e estabelece uma associação entre o crescimento da taxa geométrica das populações nos países em desenvolvimento e a demanda crescente por esses recursos, especialmente energéticos (RATTNER, 1977). Assim, é notório que o alto consumo de energia traz várias consequências para o meio ambiente. Na construção de uma hidrelétrica, por exemplo, há impactos sociais e ambientais; no represamento do rio, em uma termoeletrica, está presente a emissão de gases de efeito estufa; em uma usina nuclear há a produção de rejeitos radioativos.

Além de todas as consequências ambientais que o alto consumo de energia pode nos trazer, existe o fator econômico que impacta diretamente as famílias brasileiras. Uma família de classe média alta consome em torno de 39.600 kWh por ano, que pode chegar a um valor de até R\$1.495,00 por mês (VERRI, 2013). Como consequência muitas famílias utilizam-se de estratégias em suas rotinas como alternativa para mitigar o custo oriundo do consumo energético em suas residências, utilizando-se de métodos simples como evitar o uso do ar condicionado, manter luzes desligadas, acumular roupas para lavar todas em um dia da semana, entre outros.

Com o tempo, é de se esperar que alguns aparelhos se tornem obsoletos e que suas tecnologias fiquem ultrapassadas, podendo ocasionar em um consumo ainda maior de energia. Em consequência disso, é comum vermos campanhas de conscientização para a troca de lâmpadas tradicionais por lâmpadas de LED, famílias se preocupando com a troca de algum eletrodoméstico para reduzir o consumo de energia, programas de incentivo à eficiência energética como o da CELESC que possui até um simulador de gastos por ambiente para alertar e auxiliar o consumidor final (CELESC, 2019). A eficiência energética vem sendo uma preocupação pública: em outras palavras, sabe-se que é possível economizar energia com mudanças comportamentais dos usuários, ajudando não só financeiramente como na emissão de poluentes (ALTOÉ *et al.*, 2017).

Pensando nisso, começou-se a desenvolver um produto de telemetria residencial, juntamente com outras três alunas do IFSC, do curso de Engenharia Eletrônica, para um projeto ligado ao concurso "Ideias Inovadoras", cuja ideia consiste em um módulo eletrônico, *hardware*, o qual monitora o consumo energético em um ambiente residencial, transmitindo os dados via WI-FI para um banco de dados e um *software* de gerenciamento de energia responsável por

exibir os dados para o consumidor.

O sistema permitirá o controle de gastos por ambiente, por aparelhos ou pelo consumo geral da residência e seus dados sobre o consumo serão fornecidos em tempo real para o usuário, possibilitando uma análise remota. O produto proporcionará a redução dos gastos por gerenciamento dos aparelhos ligados à rede do usuário; assim, por exemplo, avisará quando algum equipamento está ligado à tomada e não está sendo utilizado; mostrará quando algum aparelho está obsoleto ou gastando mais do que deveria, além de analisar o consumo de cada dispositivo ligado à rede, informando dados do uso e fornecendo dicas para o melhor aproveitamento energético.

O *hardware* possui sensores para monitorar o consumo em Watts e um microcontrolador para o processamento e comunicação desses dados. No *software*, o cliente terá acesso ao consumo pontual, podendo saber os horários de picos e ainda ter acesso em tempo real quando as tomadas e as lâmpadas estão sendo utilizadas.

Para o desenvolvimento deste produto, faz-se necessária uma análise mais profunda de cada aparelho; assim, o trabalho que segue trata da análise de sinal dos aparelhos, a fim de caracterizá-lo por uma assinatura de carga, e também do reconhecimento do eletrodoméstico a partir deste sinal. Este Trabalho de Conclusão de Curso foi realizado no Curso de Engenharia Eletrônica, do Departamento Acadêmico de Eletrônica do Instituto Federal de Santa Catarina – Câmpus Florianópolis e utilizou pesquisas qualitativas, descritivas e bibliográficas em sites indexados e bases de dados que tratam do tema.

O trabalho apresenta-se estruturado da seguinte maneira: após o capítulo da introdução, será apresentada a revisão bibliográfica do tema e das tecnologias utilizadas para desenvolvimento do trabalho; em seguida, será mostrada a descrição do sistema desenvolvido, o qual apresentará um fluxograma do reconhecimento das cargas; no penúltimo capítulo, serão mostradas todas as etapas do desenvolvimento do projeto - onde será apresentado o sistema que irá fazer o reconhecimento do aparelho; por fim, no último capítulo serão descritos os resultados obtidos, problemas encontrados no decorrer do desenvolvimento e possíveis áreas para aprimoramentos e trabalhos futuros.

1.1 Justificativa

A preocupação com a redução do consumo e o uso racional da energia elétrica tem aumentado nos últimos anos devido, principalmente, a dois fatores: as mudanças na estrutura do setor elétrico brasileiro e a preocupação em reduzir os impactos ambientais com a construção de novas fontes de energia (MARTINEZ *et al.*, 2009).

Assim, percebe-se que a necessidade de economizar energia elétrica é de extrema importância no contexto em que vivemos; pensando nisso, o monitoramento e o reconhecimento das cargas elétricas presentes em uma residência podem ajudar ativamente a alcançar padrões

comportamentais corretos. Qualquer pessoa, por exemplo, poderia ser capaz de, a partir de uma análise mais aprofundada de seus aparelhos eletrônicos, verificar a razão de um possível aumento na conta de energia elétrica, a qual poderia ocorrer devido a algum problema no equipamento, obsolescência ou qualquer outro motivo e, a partir dessa descoberta, poderia fazer a troca ou manutenção de seus aparelhos, possibilitando a redução do gasto energético.

Alertar o consumidor desses dados dá o conhecimento e a oportunidade de ele tratar ou efetuar a troca de aparelhos que estão consumindo mais que o normal, ou até ter conhecimento de que alguns eletrodomésticos que permaneciam ligados à rede por um longo período de tempo, sem necessidade, correspondem a uma significativa parte de sua conta de energia. Eis a relevância desta pesquisa, que buscará contribuir com a sociedade e com o meio ambiente para a redução do consumo de energia elétrica.

1.2 Definições do problema

No desenvolvimento desta pesquisa, foi questionado como iria ser feito o reconhecimento de cada aparelho ligado à rede através das assinaturas de cada carga. Cada aparelho possui uma assinatura de energia exclusiva que permite distingui-lo dos outros. Essas assinaturas podem ser observadas nas formas de onda de tensão e corrente fornecidas ao dispositivo, ou em reproduções processadas desses sinais, como a energia real fornecida e a energia reativa ou harmônicas (CHANG; LIN; YANG, 2008). Levando-se em conta esse contexto, esta pesquisa questiona: como reconhecer os aparelhos ligados à rede?

1.3 Objetivo Geral

Compreender parâmetros que, juntos, são capazes de formar uma assinatura única para cada tipo de aparelho eletrônico e, com isso, reconhecer esse aparelho através de sua carga com algoritmos de inteligência desenvolvidos por meio de linguagem de programação.

1.4 Objetivos Específicos

O trabalho tem como objetivos específicos:

- a) identificar parâmetros relevantes no estudo das cargas;
- b) reconhecer cargas ligadas à rede;
- c) definir uma metodologia eficiente para o reconhecimento de assinaturas de cargas.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo trata de todo embasamento do trabalho, toda a teoria estudada anteriormente que deu origem ao tema e ao desenvolvimento desta monografia. Para proporcionar o avanço em um campo do conhecimento é preciso primeiro conhecer o que já foi realizado por outros pesquisadores e quais são as fronteiras do conhecimento naquela área (VIANNA, 2001).

2.1 Energia elétrica

A energia elétrica, baseada na geração de diferença de potencial elétrico em dois pontos, a qual permite uma corrente elétrica entre ambos, é uma fonte de energia proveniente da transformação de fontes primárias disponíveis no planeta (STOODI, 2020). Essas fontes podem ser de origem renovável ou não renovável; dentre as renováveis temos: energia solar, eólica, hidrelétrica e de biomassa. E dentre as energias não renováveis encontramos: combustíveis fósseis e nuclear.

A energia elétrica é indispensável para a maioria dos aparelhos eletrodomésticos que encontramos hoje nas residências, ela tem vantagens de ser facilmente transportada e apresentar uma baixa perda energética.

Sendo primordial para humanidade, hoje em dia, a energia elétrica vem sendo cada vez mais utilizada pela população elevando o consumo deste recurso e gerando preocupações ambientais para o planeta e, pessoais, para as famílias que a utilizam frequentemente.

2.2 Eficiência Energética

Muito se tem falado de eficiência energética, mas o que é isso? Eficiência energética é o uso racional de energia, consiste em utilizar este recurso de maneira mais eficiente.

A Eficiência energética consiste na forma com que desempenhamos um serviço ou atividade utilizando a menor quantidade de energia possível, ou seja, consiste na relação entre a quantidade de energia empregada e aquela disponibilizada para a sua realização (DUARTE, 2016).

Uma pesquisa sobre a eficiência energética, disponibilizada pela ACEEE (American Council for an Energy-Efficient Economy), em 2014, analisou o consumo de energia em 16 países; o Brasil ficou na posição 15ª posição; em primeiro lugar, estava a Alemanha, país que se destaca pelo uso de seus recursos (ECONOMY, 2014). Segundo a Energia (2015), se utilizássemos equipamentos mais eficientes e as instalações elétricas fossem melhor dimensionadas, alcançaríamos uma economia de 8,8% no consumo de energia elétrica. Esse resultado nos leva a pensar que mudanças comportamentais são extremamente importantes para uma melhor eficiência energética.

Existem diversas maneiras de economizar energia além da troca de instalações elétricas e equipamentos. O uso adequado de cada eletrodoméstico também interfere no consumo energético de uma residência, assim como também entender os custos e picos de energia de cada aparelho de um ambiente pode ajudar a identificar gastos desnecessários com a energia ou até aparelhos que já estão obsoletos e não haviam sido identificados.

2.3 Sistemas de Monitoramento

Para garantir uma eficiência energética é preciso analisar melhor o consumo, sendo necessário um sistema de monitoramento, as assinaturas de cargas podem ser adquiridas basicamente de duas formas: pelo monitoramento de carga não intrusiva, NILM, ou pelo monitoramento de carga intrusiva, ILM.

No monitoramento de carga não intrusiva, temos um único medidor que é capaz de monitorar e medir o consumo da casa inteira, ou seja, um módulo é ligado ao painel de energia da casa e este é capaz de diferenciar as cargas que estão sendo ligadas desacoplando-as e distinguindo-as das outras ligadas simultaneamente.

2.4 Identificação de carga não intrusiva

Para que o monitoramento das cargas seja eficaz é necessário que seja feita a identificação das mesmas. De acordo com Hoyo-Montañó *et al.* (2018), a classificação dos aparelhos eletrônicos pode ser dividida em quatro categorias distintas, são elas:

- a) onde existem apenas dois estados operacionais, ligado e desligado;
- b) onde há vários níveis de consumo definidos e um funcionamento cíclico;
- c) onde há um número infinito de pontos de operação quando ligados;
- d) onde operam durante longos períodos de tempo.

A Equação 1 indica onde há o consumo total de energia agregado a dispositivos operando dentro de uma instalação elétrica:

$$P(t) = \sum a_i(t) * P_i + e(t) \quad (1)$$

Onde:

$P(t)$ = consumo de energia ao longo do tempo

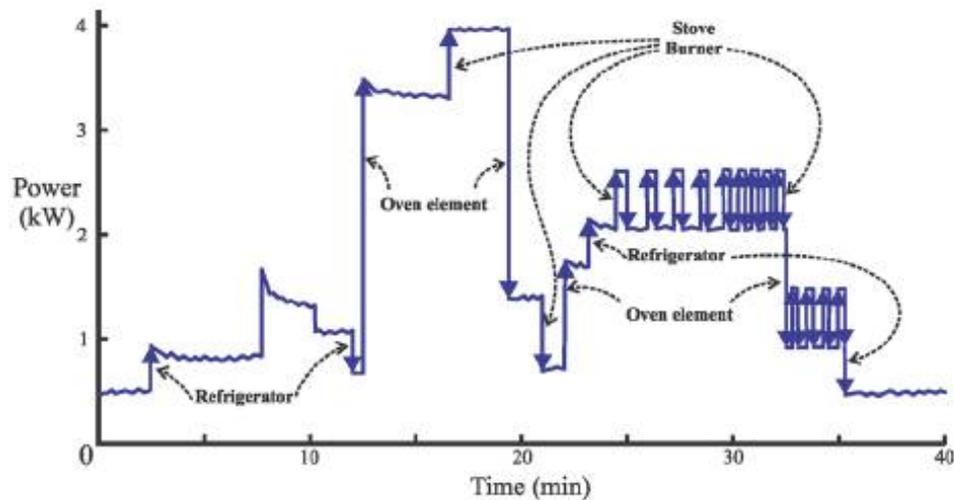
$a_i(t)$ = vetor de ativação do dispositivo i , com valores 0 e 1 quando o dispositivo está desligado ou ligado, respectivamente, durante um período de tempo t

P_i = vetor de potência do dispositivo i

$e(t)$ = erro ou ruído

Na Figura 1, observamos o consumo agregado a vários dispositivos distintos.

Figura 1 – Consumo de energia agregado a vários dispositivos distintos



Fonte: Hart (1992).

2.5 Medição de Corrente

Para o desenvolvimento das assinaturas de cargas, é necessário gerar dados por meio da medição de corrente e tensão dos eletrodomésticos envolvidos. São utilizados valores de corrente eficaz para o reconhecimento dos aparelhos, essa pode ser representada pelo valor da raiz quadrada da média do quadrado como na Equação 2 (NAKASHIMA, 2007).

$$I_{rms} = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} i^2(t) dt} \quad (2)$$

O valor eficaz ou RMS de corrente, ou tensão, é relativo ao calor dissipado em uma resistência, ele representa o valor de uma corrente, ou tensão, contínua que gera uma dissipação de potência igual a corrente, ou tensão, periódica (NAKASHIMA, 2007).

Para medir a corrente, existem vários métodos e sensores já conhecidos, podemos ver alguns a seguir.

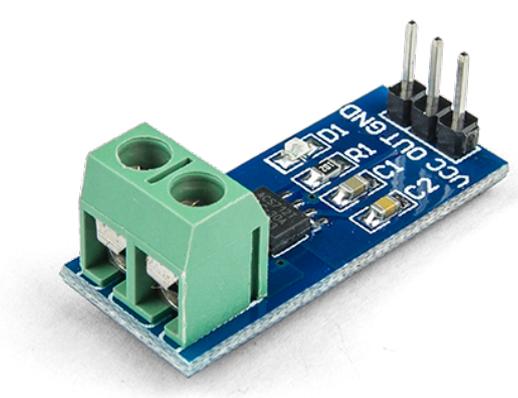
2.5.1 Efeito Hall

Segundo Endalécio (2013), o Efeito Hall se dá nas regiões com diferença de potencial, formadas pela presença de um campo magnético, que atua nas cargas que circulam pelo condutor, codificando a trajetória destas.

Através da concentração de cargas é possível determinar o sentido que a corrente circula pelo condutor, observando o sinal da diferença de potencial gerado através da presença do campo magnético perpendicular a direção da corrente (ENDALÉCIO, 2013).

Como exemplo desta tecnologia temos o ACS712, um sensor de corrente que trabalha com medições de -30 A a 30 A, com uma alimentação de 5 V e uma sensibilidade da saída de 66m V/A, podendo ser utilizado com cargas AC ou DC (ROBOCORE, 2018). Esse sensor é instalado em série com o circuito a ser medido, na Figura 2 podemos observar o sensor em questão.

Figura 2 – Sensor ACS712



Fonte: ROBOCORE (2018).

A máxima leitura que podemos ver na saída analógica desse sensor é de 4,48 V, para sua máxima corrente de 30 A, uma vez que o sensor varia de 0,066 V a cada 1 A. Levando isso em conta, podemos dizer que o sensor não é recomendável para correntes baixas, pois a leitura não será muito precisa (ROBOCORE, 2018).

2.5.2 Transformador de Corrente

O transformador de corrente (TC) é um mecanismo monofásico que tem dois enrolamentos isolados um do outro, mas, acoplados magneticamente, e que são utilizados para reduzir o valor da corrente, promovendo segurança, isolando eletricamente o circuito de potência e padronizando os valores de corrente (MARDEGAN, 2010).

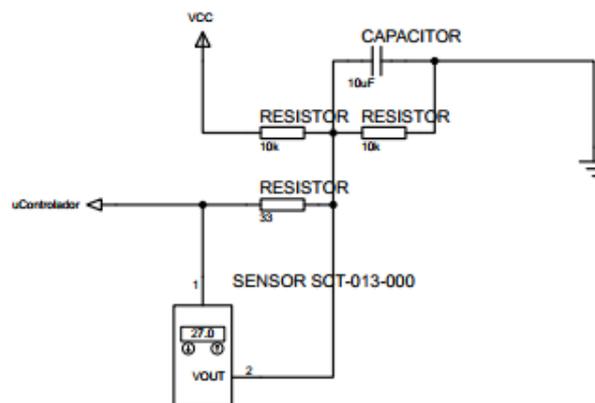
Para o monitoramento de cargas elétricas é comum vermos a utilização desses sensores, por ser um sensor não intrusivo, Hoyo-Montaña *et al.* (2018) cita que utilizou esse tipo de sensor para facilitar a instalação do aparelho e baratear o custo do mesmo.

Segundo Bacurau *et al.* (2014), sensores do tipo TC requer um tipo de filtro *anti-aliasing* mais simples do que o *shunt*, é mais fácil de instalar e provê isolamento elétrica para o circuito de medição.

Como exemplo dessa tecnologia temos o sensor SCT-013-000, que é bem recomendado, por ser um sensor de corrente muito conhecido no mercado e bastante utilizado. Um sensor não invasivo, que permite correntes até 100 A; para a utilização desse sensor, é necessário incluir junto a ele um pequeno circuito para condicionar o sinal antes que ele seja enviado ao controlador.

Para condicionar a corrente o circuito proposto é composto por dois resistores de 10k Ω , um resistor de 33 Ω e um capacitor de 10 μF . Eles são ligados conforme a Figura 3.

Figura 3 – Circuito para o sensor de corrente



Fonte: Acervo Pessoal (2020).

2.5.3 Resistência *Shunt*

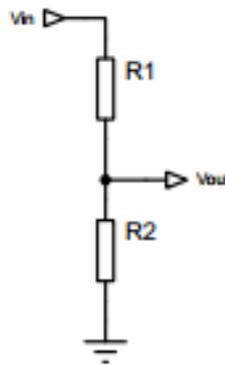
De acordo com Rangel-Abundis (2006), o *Shunt* é um resistor que é ligado em série com a carga e por meio da queda de tensão é possível chegar ao valor da corrente.

De acordo com Bacurau *et al.* (2014), *Shunt* não é o melhor método para medir corrente nesse tipo de aplicação, por precisar de um filtro *anti-aliasing* mais complexo e por acrescentar custos de instalação. Nesse mesmo artigo é utilizado esse tipo de sensor como forma alternativa para a medição, porém ele se apresentou bastante ruidoso.

2.6 Divisor de Tensão

A tensão da rede é muito alta para ser medida diretamente pelo microcontrolador; logo é necessária uma transformação para um nível de tensão seguro. Isso pode ser obtido por um simples circuito de divisor de tensão, o qual consiste em criar uma tensão elétrica que seja proporcional à outra tensão, conforme é mostrado na Figura 4.

Figura 4 – Circuito divisor de tensão



Fonte: Acervo Pessoal (2020).

A tensão de saída pode ser encontrada utilizando a Equação, 3 , a seguir.

$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2} \quad (3)$$

Essa equação serve para quando o divisor não possui carga; assim, admitimos que a resistência da carga é infinita e toda a corrente que passa por R1 vai para R2. Caso o divisor possua uma carga, a mesma precisa ser considerada.

Em circuitos elétricos podemos obter o valor eficaz da tensão com a raiz quadrada do valor médio do quadrado da função, como na Equação 4 (OLIVEIRA; LIMONGI; CHAVES, 2010).

$$V_{rms} = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} v^2(t) dt} \quad (4)$$

O valor médio da corrente pode ser obtido através da Equação 5, pela sua integral em um intervalo dividido pelo seu período, em geral, o valor CC é utilizado para caracterizar correntes e tensões que não mudam de sinal ao longo do tempo (OLIVEIRA; LIMONGI; CHAVES, 2010).

$$V_{medio} = \frac{1}{2\pi} \int_0^{2\pi} V_m \cos(\omega t) dt \quad (5)$$

2.7 Aquisição da Potência Ativa e Reativa

Além dos dados de tensão e corrente se faz necessário obter valores para potência ativa e potência reativa. Esses valores podem ser obtidos através de cálculos com os dados de corrente e tensão eficaz.

A potência ativa pode ser obtida através da multiplicação da tensão e da corrente no instante, como na Equação 6. Ao adquirir os dados para corrente e tensão é possível calcular a potência ativa do aparelho ao longo do tempo.

$$P = V * I \quad (6)$$

Também torna-se possível calcular a potência reativa do eletrodoméstico através destes dados, porém para chegar em um resultado para a potência reativa é necessário obter a potência aparente, Equação 7. Após obter a potência aparente e a potência ativa podemos chegar em um resultado para a potência reativa através da Equação 8.

$$|S| = \hat{V} * \hat{I} \quad (7)$$

$$Q = \sqrt{s^2 - P^2} \quad (8)$$

2.8 Taxa de Distorção Harmônica

A taxa de distorção harmônica é calculada com base no espalhamento espectral, tendo em vista a energia da frequência e das componentes harmônicas de ordem superior (NASCIMENTO; BERGER; ROCHA, 2007).

Segundo Bacurau *et al.* (2014) não há registros de sistemas comerciais que utilizem componentes harmônicos em sua assinatura de carga, apesar da potencialidade dessa abordagem.

2.9 EVM430-F6736

Atualmente existem *hardwares* capazes de capturar esses dados com maior facilidade, um deles é o EVM430-F6736, Figura 5. Apesar de o desenvolvimento deste trabalho tomar como base dados já pré coletados e armazenados em um banco de dados, o intuito do projeto total é capturar os dados através do *hardware* EVM430-F6736, para fazer análise em tempo real dos eletrodomésticos do usuário. Ele é um módulo medidor de eletricidade monofásico baseado no controlador MSP430F6736. O módulo pode ser conectado às linhas de forças principais e tem entradas para tensão, corrente e uma terceira conexão para configurar anti-violção (Texas Instruments, 2015a). O EVM430-F6736 possui um *software* programável e facilmente adaptado para diversas necessidades e estruturas, permite que o usuário execute aplicativos de medição de eletricidade em tempo real (Texas Instruments, 2015a).

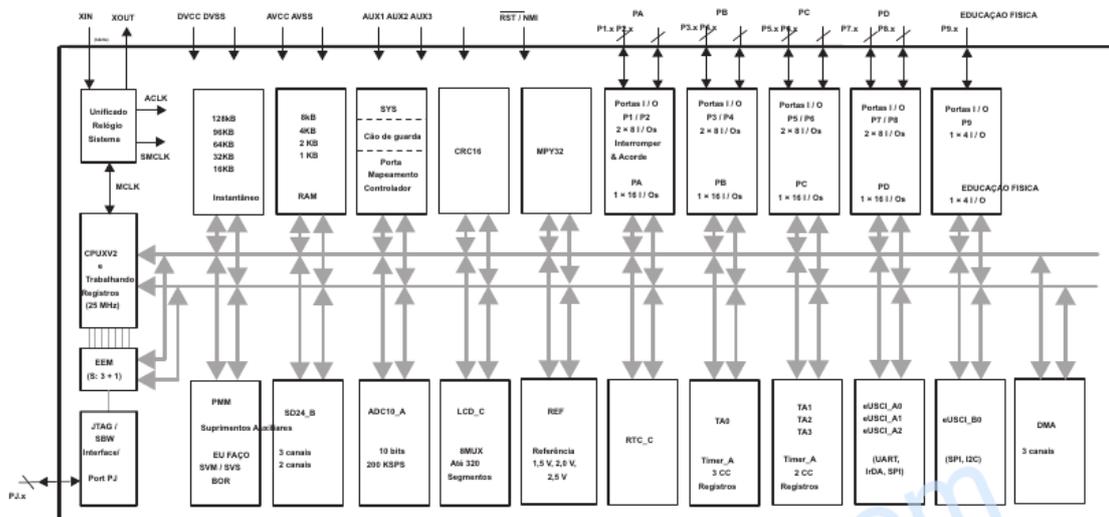
Figura 5 – EVM430-F6736



Fonte: (Texas Instruments, 2015a)

O MSP430F6736, presente no EVM430-F6736, é um microcontrolador de 16 bits utilizado para a medição e detecção de energia de baixo custo, seu diagrama de blocos pode ser visualizado na Figura 6. Ele é capaz de despertar tempos abaixo de 1μ segundo, permitindo, assim, que o microcontrolador permaneça no modo de espera por mais tempo e reduzindo o consumo médio de corrente (Texas Instruments, 2015b). O microcontrolador é composto por vários periféricos como; oscilador interno, temporizador, USART, SPI, I2C, entre outros.

Figura 6 – Diagrama de blocos MSP430F673X



Fonte: (Texas Instruments, 2015a)

O microcontrolador MSP430 é próprio para este tipo de aplicação, além de possuir um custo baixo. O MSP430 tem uma taxa de amostragem de 200 kHz, o esperado para a aplicação é cerca de 120 kHz (BACURAU *et al.*, 2014).

2.10 Processamento de Sinais

Após captar os dados é importante tratá-los para extrair as informações necessárias para o reconhecimento da carga, para isso podemos utilizar do processamento de sinais, que é caracterizado por processar informações extraídas a fim de obter respostas para variáveis de algum estado distante da natureza envolvida (ALCALM; OLIVEIRA, 2011).

2.10.1 Processamento analógico de sinal

O processamento analógico de sinal é feito através de componentes analógicos, como resistores, capacitores e indutores. Esses componentes podem sofrer variações de acordo com a temperatura, vibrações mecânicas, variações de tensão, entre outros fatores que podem influenciar drasticamente na eficiência destes circuitos (PELLENZ, 2005).

2.10.2 Processamento digital de sinal

Com o processamento digital de sinais podemos atingir, na grande maioria das vezes, resultados melhores que no processamento analógico de sinal; isso se dá uma vez que as técnicas utilizadas hoje em dia são extremamente eficazes. Utilizamos o processamento digital de sinais diariamente em vários aparelhos, tais como, celulares, tablets, computadores, automóveis, etc.

De acordo com Puhlmann (2020), "O processamento digital de sinais propriamente dito, é uma tecnologia, uma ciência, uma série de conceitos abstratos que se traduz na aplicação de algoritmos computacionais para a realização de operações específicas sobre dados digitais".

Através do processamento digital de sinal é simples alterar qualquer dado ou parâmetro em sua aplicação. Ao adicionar o processamento digital de sinal, é reduzida a suscetibilidade ao ruído, custo, consumo de potência e tempo de desenvolvimento (PELLENZ, 2005).

2.11 Assinatura de Cargas

Com os dados extraídos e tratados é possível desenvolver as assinaturas de cargas, um conjunto de características elétricas únicas, utilizadas para identificação de aparelhos que estão ligados à rede (BACURAU *et al.*, 2014). Elas podem ser compostas por diversos parâmetros elétricos, como, tensão, corrente, potências ativa e aparente, harmônicas, dentre outros.

Para gerar a assinatura de carga primeiro é preciso medir a corrente e a tensão da residência, depois calcular o consumo acumulado, a potência ativa, a potência aparente e o fator de potência. A assinatura de carga é desenvolvida para cada aparelho, uma vez que o mesmo é ligado sozinho, para que seja possível obter os parâmetros isolados para cada eletrodoméstico a ser estudado. Os dados de cada aparelho são coletados e salvos em um banco de dados para uma pré-classificação, realização de treinamento e de teste (SOUSA *et al.*,).

De acordo com Bacurau *et al.* (2014), as assinaturas de cargas para a identificação de aparelhos, que estão ligados à rede, podem ser definidas de várias maneiras, sendo a mais comum através da curva de potência ativa que o aparelho consome ao longo do tempo. Essas podem ser apresentadas basicamente de duas maneiras, no domínio do tempo ou no domínio da frequência. O algoritmo para o reconhecimento das cargas pode funcionar levando em conta o estado estacionário, quando a carga se encontra em regime, ou o estado transiente, quando a carga é ligada a rede, ou até com uma combinação das duas formas.

Conforme esses mesmos autores (2014), chamamos de eventos as mudanças no estado de operação do aparelho, os quais correspondem à carga sendo ligada ou desligada e podem ser caracterizadas em magnitude e sinal. Estes eventos podem nos ajudar a identificar as cargas que estão ligadas à rede. Segundo Bacurau *et al.* (2014) para a discriminação de cargas e processamento de parâmetros elétricos constantes, é necessário o uso de algoritmos.

Para o desenvolvimento deste trabalho, foi utilizado um banco de dados já populado com parâmetros e assinaturas de eletrodomésticos extraídos anteriormente pela iCoSys, Instituto de Sistemas Complexos. Posteriormente, será utilizado um algoritmo de inteligência artificial para a identificação do aparelho.

2.12 Banco de Dados

As Assinaturas de cargas utilizadas na aplicação são salvas em um banco de dados que permite a busca e leitura dos parâmetros armazenados.

De acordo com Silberschatz *et al.* (1997), banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico, ou seja, é um agrupamento de informações que se relacionam e abordam um mesmo assunto.

Um sistema de banco de dados é um conjunto de quatro componentes básicos: dados, *hardware*, *software* e usuários. Tem como objetivo isolar o usuário de dados brutos que ele não precisa visualizar, abstraindo-os para dados tratados e apresentados de forma amigável ao usuário final do sistema em questão; o banco de dados é utilizado para armazenar os dados e apresentar apenas o necessário conforme a aplicação.

Para desenvolver este trabalho, foi utilizado o ACS-F1, que é um banco de dados de assinaturas de consumo de aparelhos aplicado para o reconhecimento de cargas ligadas à rede. O banco foi desenvolvido em 2013, por meio de sensores *low-end*, baseados em plugues que

medem o consumo elétrico em baixa frequência, geralmente a cada 10 segundos, onde foram realizadas duas sessões de aquisição de uma hora em mais ou menos 100 eletrodomésticos classificados em 10 categorias e 6 medidas de consumo. O acesso a esse banco de dados é gratuito e está disponibilizado pela iCoSys em seu próprio site.

Os proprietários do banco de dados ACS-F1 são o Instituto iCoSys, Universidade de Ciências Aplicadas HES-SO//Fribourg, Faculdade de Engenharia e Arquitetura, Pérolles 80, CH-1700 Fribourg, Suíça.

2.13 Aprendizagem de Máquina

Para o reconhecimento das assinaturas de cargas através do banco de dados é necessário utilizar do aprendizado de máquina, ou *machine learning*, que é o estudo de códigos computacionais que se aprimoram com o passar do tempo automaticamente, à medida que vai ganhando experiência.

O aprendizado de máquina aborda a questão de como construir computadores que melhoram automaticamente com a experiência. É um dos campos técnicos de mais rápido crescimento da atualidade, situado na interseção da ciência da computação e estatística e no núcleo da inteligência artificial e da ciência de dados. O progresso recente no aprendizado de máquina foi impulsionado pelo desenvolvimento de novos algoritmos e teoria de aprendizado e pela explosão contínua na disponibilidade de dados online e computação de baixo custo. A adoção de métodos de aprendizado de máquina com uso intensivo de dados pode ser encontrada em toda a ciência, tecnologia e comércio, levando a mais tomadas de decisões baseadas em evidências em muitas esferas da vida, incluindo assistência médica, fabricação, educação, modelagem financeira, policiamento e marketing (JORDAN; MITCHELL, 2015).

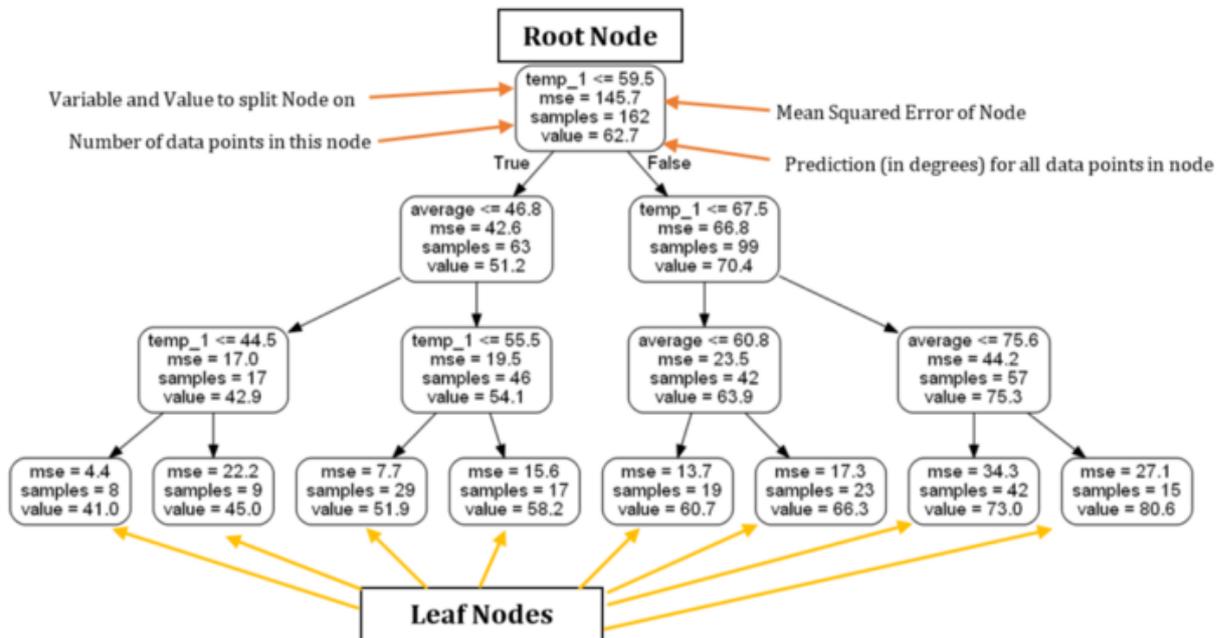
A aprendizagem de máquina é um segmento de pesquisa da Inteligência Artificial, em que são criados e usados modelos que são provenientes de dados históricos (SOUSA *et al.*,).

Segundo Brownlee (2020), os algoritmos de aprendizagem de máquina são utilizados para gerar modelos classificadores formando um conjunto de exemplos.

2.13.1 Árvore de Decisão

Segundo Campos (2017), árvore de decisão é um método de aprendizado de máquinas supervisionado não-paramétrico; são frequentemente utilizados em tarefas de classificação e regressão. Ou seja, é uma forma de fluxograma, em que existem nós, alternativas, possíveis cenários e término. As regras de uma decisão são armazenadas nos nós da árvore e os resultados são representados nas suas folhas, como podemos ver na Figura 7.

Figura 7 – Imagem ilustrativa de uma árvore de decisão



Fonte: Koehrsen (2017).

A construção automática de uma árvore de decisão a partir de um conjunto de dados é considerada aprendizado de máquina. De acordo com Lucidchart (2018), uma árvore de decisão possibilita comparar possíveis ações com base em seus custos, probabilidades e benefícios e é utilizada para prever a melhor escolha, matematicamente. Ela agrega consigo várias vantagens, dentre elas, a facilidade de compreensão. Para ser utilizada no aprendizado de máquina, o método considera as observações sobre algo para prever seu valor. Neste tipo de árvore, são representados dados, e não decisões. Para a árvore ser classificada como ótima, ela precisa ter o maior número de dados representados com o menor número de níveis possível.

Para a construção de uma árvore de decisão é denominado os atributos que serão utilizados nas decisões, em cada nó da árvore, em seguida calcula-se o ganho de informação para aquela decisão, o ganho de informação é dado a partir do cálculo de entropia do dado, quão puro é aquele dado. Com o ganho de informação temos uma representação do quanto aquela decisão é capaz de agregar no resultado final, então a árvore é organizada com as decisões que tem um maior ganho de informação mais ao topo da árvore, pois ela será capaz de segregar o maior número de dados.

Para exemplificar a construção da árvore de decisão foi construído uma tabela ilustrativa, Tabela 1, nela podemos observar dados que formarão nossas "decisões", os nós da nossa árvore.

Tabela 1 – Tabela ilustrativa

Eletrô	Pot. Ativa média (W)	Corrente Média(A)
microwave	199.62	0.97
lamp	7.86	0.03
tv Panasonic	18.30	0.24

Fonte: Acervo Pessoal (2020).

Para construir a árvore, à fim de distinguir o aparelho televisão, calculamos o ganho de informação para cada uma das decisões; potência média > 100 W; corrente média > 0,1 A. Para obter o ganho de informação é necessário calcular a entropia através da Equação 9.

$$Entropia(S) = - \sum_{i=1}^n p_i * \log_2 p_i \quad (9)$$

A equação da entropia é baseado na probabilidade, como temos um total de três aparelhos e apenas um é um televisor a probabilidade é de 1/3, adicionando os dados na equação obtemos o resultado representado na equação 10.

$$Entropia(S) = -(1/3 * \log_2 1/3 + 2/3 * \log_2 2/3) \quad (10)$$

Resolvendo a equação temos que a entropia total do sistema é igual a 0,93. Repetindo o mesmo processo para calcular a entropia de cada uma das decisões temos que a entropia do ramo "potência média > 100 W" é de 0 e do ramo "potência média < 100 W" é de 1, já a entropia dos ramos "corrente média" é de 0. Posteriormente é calculado o peso de cada um dos ramos pela divisão entre o número de amostras daquele ramo e o número de amostras do pai. Tendo esses dados calculados é possível obter o ganho de informação de cada uma das decisões através da Equação 11.

$$Ganho(S, A) = Entropia(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} * Entropia(S_v) \quad (11)$$

Adicionando os dados para a decisão "potência média > 100 W", temos que o ganho de informação obtido é de 0,26, replicando o processo para a decisão "corrente média > 0,1 A" temos um ganho de informação de 0,93. Com isso podemos admitir que a decisão envolvendo corrente nesse caso estaria mais ao topo da nossa árvore por apresentar um maior valor para o ganho de informação.

2.13.2 Naive Bayes

Naive Bayes é um classificador probabilístico, utilizado para selecionar a melhor hipótese de acordo com os dados. O teorema de Bayes nos fornece um método para calcular a

probabilidade de uma hipótese referente a um dado inicial, ele é representado pela equação 12.

$$P(h|d) = \frac{P(d|h) * P(h)}{P(d)} \quad (12)$$

Onde:

$P(h|d)$ = probabilidade posterior

$P(d|h)$ = probabilidade dos dados d , considerando a hipótese h sendo verdadeira

$P(h)$ = probabilidade da hipótese h ser verdadeira

$P(d)$ = probabilidade dos dados

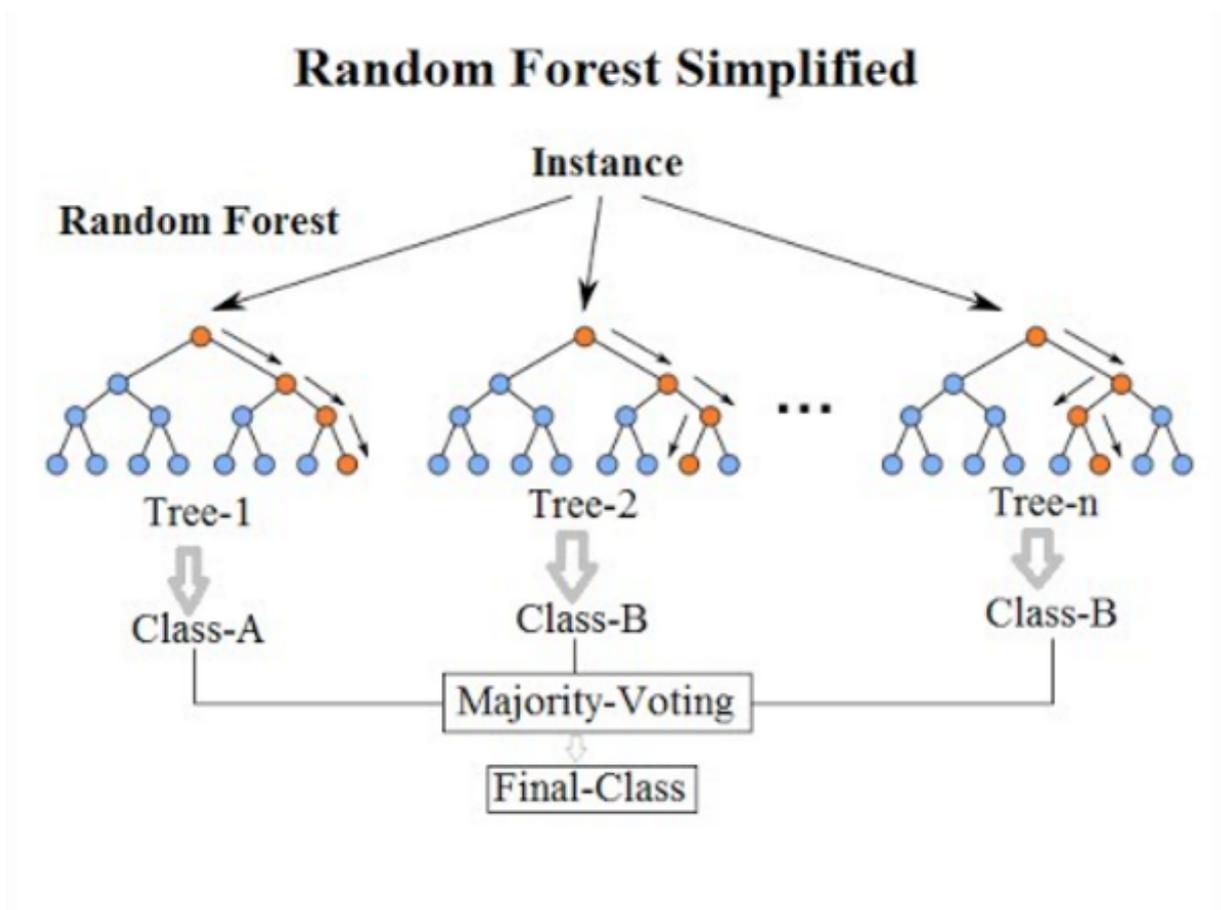
Naive Bayes é um algoritmo de classificação para problemas de classificação binária e multi-classe. A técnica é mais compreensível quando descrita usando valores de entrada binários ou categóricos (BROWNLEE, 2016).

2.13.3 *Random Forest*

A *Random Forest* é um classificador aleatório baseado em árvores de decisão, que consiste em uma combinação de classificadores de árvore de decisão, em que cada classificador é gerado a partir de um vetor aleatório amostrado independentemente do vetor de entrada (BREIMAN, 1996).

A *Random Forest* ou floresta aleatória é composta por N árvores de decisão que são geradas aleatoriamente, diferentes uma das outras, elas funcionam em paralelo e cada uma obtém um resultado, que no final dará o resultado da aplicação, Figura 8.

Figura 8 – Imagem ilustrativa de uma floresta aleatória



Fonte: Junior e Covolan (2018).

Para selecionar os atributos utilizados na indução de uma *Random Forest* usamos o índice de Gini, que é responsável por medir a impureza de um atributo em relação às classes (PAL, 2005). O índice de Gini pode ser representado pela Equação 13:

$$Entropia(S) = \sum_{j \neq i} \sum (f(C_i, T)/|T|)(f(C_j, T)/|T|) \quad (13)$$

Onde:

$f(C_i, T)/|T|$ = probabilidade do caso selecionado pertencer a classe C_i

T = conjunto de treinamento

C_i = classe

Para desenvolver uma floresta aleatória, é necessário decidir o número de recursos que serão utilizados em cada nó, para gerar a árvore, e também decidir a quantidade de árvores que serão utilizadas na classificação.

Então é necessário extrair os dados e separá-los em recursos e destinos, onde recursos serão os dados que vão ser utilizados para fazer a classificação e destinos os dados que serão classificados. Após separá-los é denominado um alvo, que será o dado classificado e em seguida é dividido o dados em conjunto de treinamento e em conjunto de teste e treinamos o modelo. O conjunto de treinamento será utilizado para treinar a floresta aleatória, então ela fará a classificação tendo em vista o resultado esperado. Já o conjunto de teste será utilizado para testar a aplicação, ele fará a classificação sem saber o resultado que se espera. Assim conseguimos calcular a precisão da resposta.

Para classificar um novo conjunto de dados, os conjuntos são passados para cada uma das N árvores e a floresta decide uma classe com o maior número de "votos" para esse caso (PAL, 2005).

2.14 Linguagens de Programação

Para desenvolver uma aplicação em *software* é necessário escolher uma linguagem de programação, para fazer tal escolha é importante levarmos em consideração bibliotecas, adequações em sistemas operacionais, facilidade na comunicação externa, curva de aprendizado, desgaste a manutenção do código, entre outros fatores.

2.14.1 Python

De acordo com Dyck *et al.* (2016), Python é uma linguagem interessante para utilizar no tratamento desses dados por apresentar uma quantidade expressiva de bibliotecas e documentações, que ajudam no desenvolvimento do código. No artigo de Hoyo-Montaño *et al.* (2018) também é utilizado do Python para a identificação de cargas elétricas não intrusivas.

Python é uma linguagem de programação que permite integrar sistemas de forma mais eficaz e otimizar o trabalho. É uma linguagem muito utilizada principalmente para aprendizado de máquina, automação de sistemas e desenvolvimento Web, por esse motivo é uma linguagem bem conceituada no reconhecimento de cargas.

É caracterizada por ser uma linguagem de fácil aprendizagem e utilização; além disso, ela é muito adotada e utilizada no meio do desenvolvimento, apesar de ser uma linguagem antiga ela continua avançando e evoluindo.

Há uma grande gama de bibliotecas para ciência de dados ou aprendizado de máquinas que possuem interfaces Python, tornando o idioma a interface de comando de alto nível mais habitual para bibliotecas de aprendizado de máquinas e outros algoritmos numéricos (YEGULALP, 2019).

3 METODOLOGIA

Definir a metodologia de uma pesquisa é como classificá-la e, para o melhor entendimento da metodologia do trabalho; entendemos que pesquisa pode ser definida como:

... um procedimento formal com método de pensamento reflexivo que requer um tratamento científico e se constitui no caminho para se conhecer a realidade ou para descobrir verdades parciais. Significa muito mais do que apenas procurar a verdade: é encontrar respostas para questões propostas, utilizando métodos científicos. Especificamente é um procedimento reflexivo sistemático, controlado e crítico, que permite descobrir novos fatos ou dados, relações ou leis, em qualquer campo de conhecimento (LAKATOS; MARCONI, 2001)

As pesquisas desenvolvidas neste trabalho são de natureza qualitativa, descritiva e bibliográfica.

Uma pesquisa qualitativa é caracterizada por partir de estudos amplos, que vão se definindo no decorrer do desenvolvimento da pesquisa. Ela não procura empregar instrumental estatístico na análise de dados, nem medir eventos estudados e envolve o contato direto do pesquisador com a situação estudada (GODOY, 1995).

Esta pesquisa se caracteriza qualitativa, pois foi mantida uma relação direta com o tema, assim como com o problema que o trabalho se propõe a resolver, além disso, o trabalho começa com um estudo abrangente do reconhecimento de carga e o método é definido no decorrer da pesquisa.

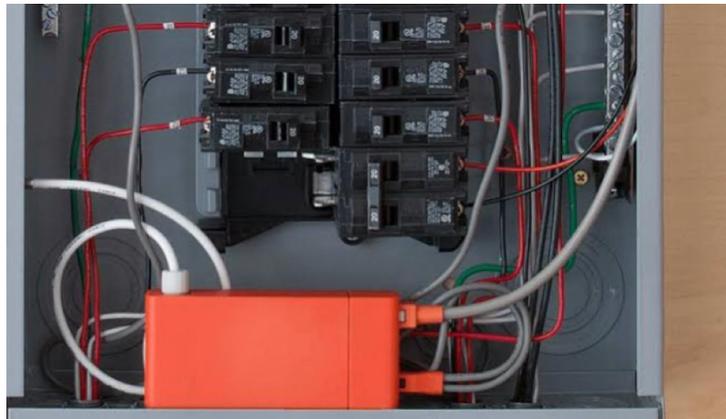
Quanto aos objetivos, esta pesquisa se classifica como descritiva, ou seja, foca em descrever as características de certo fenômeno, estabelecendo relações entre as variáveis. Envolve técnicas de coleta de dados padronizados e assume uma forma de levantamento (NASCIMENTO, 2016). O trabalho propõe descrever as etapas para a escolha do método utilizado para o reconhecimento da carga, além dos resultados e problemas encontrados.

Quanto às fontes de informação, esta pesquisa possui um caráter bibliográfico, cuja a base teórica para o estudo foi extraída de artigos, livros, reportagens, teses e dissertações em sites indexados por uma leitura seletiva, analítica e interpretativa.

4 DESCRIÇÃO DO SISTEMA

O sistema completo que engloba o tema abordado neste TCC é um módulo único ligado ao painel elétrico da residência do usuário, fornecendo os dados em tempo real através de um *software* intuitivo e inteligente, Figura 9. Para captar as medidas necessárias para a distinção dos aparelhos, foi utilizado um *hardware* composto por dois sensores de corrente, captando a corrente e a tensão presentes na rede, para que fosse possível calcular parâmetros como potência ativa, aparente e harmônicas. Essas características foram armazenadas em um banco de dados e utilizadas para o desenvolvimento das assinaturas de cargas e posteriormente para a diferenciação dos aparelhos. A captação dos dados é feita através do *hardware* EVM430-F6736, que conta com o microcontrolador MSP430 em seu sistema.

Figura 9 – Ilustração do produto instalado

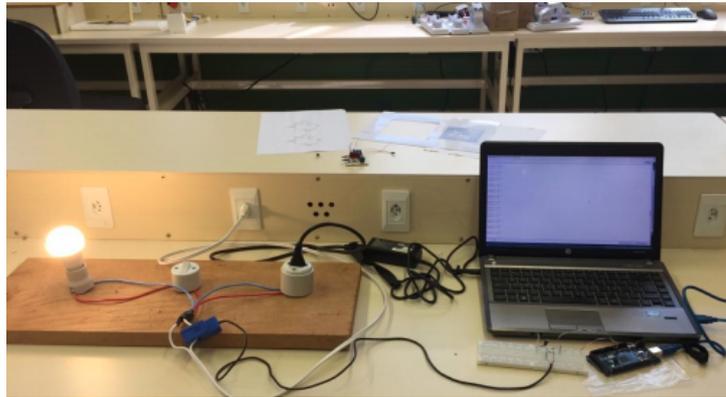


Fonte: (SENSE, 2016)

O MSP430 atende a aplicação pois além de ser próprio para medição e detecção de energia, ele tem um baixo custo e apresenta uma taxa de amostragem de 200 kHz, que seria suficiente, pois é necessário uma taxa de amostragem de mais ou menos 120 kHz (BACURAU *et al.*, 2014). Ele será utilizado para compor o módulo de medição que irá fazer a aquisição dos sinais para compor a assinatura de carga do aparelho.

Inicialmente, foram feitos testes de captação de sinal com o auxílio de um sensor de corrente SCT-013-000 e um AT-MEGA 328p; o AT-MEGA 328p apresentou testes com resultados pouco satisfatórios, pois não foi tratado nenhum tipo de ruído que pudesse atrapalhar na amostragem, além do baixa taxa de aquisição. Na Figura 10, podemos ver a construção dos primeiros testes. Posteriormente optamos por continuar o desenvolvimento do produto com o *hardware* EVM430-F6736; os testes com esse novo dispositivo ainda não foram executados por impossibilidade de utilização dos laboratórios durante a pandemia.

Figura 10 – Primeiros testes na captura do sinal na bancada



Fonte: Acervo Pessoal (2019).

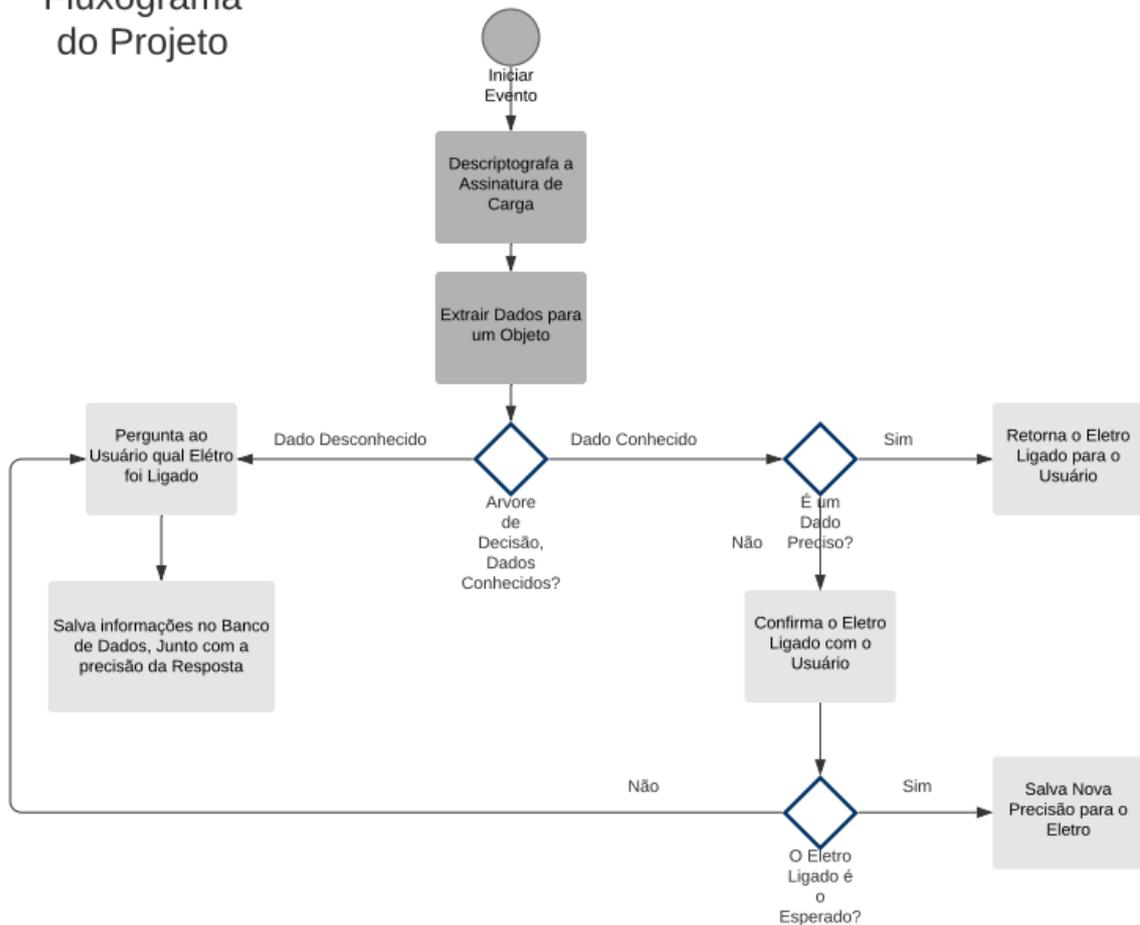
Para os testes iniciais, foram utilizados os seguintes itens: lâmpadas, alisador de cabelo, notebooks, entre outros. Apesar de não estar no cenário ideal e com resposta insatisfatória, foi possível observar, desde o primeiro momento, diferenças significativas entre os sinais de corrente dos aparelhos, que eram plotados no osciloscópio e que posteriormente, juntamente de outros parâmetros tratados, caracterizariam assinatura de carga.

Cada aparelho contém uma assinatura distinta, previamente presente no banco de dados; à medida que o produto for ganhando novos usuários, os algoritmos irão coletar mais assinaturas, populando o banco e deixando o produto cada vez mais completo.

A parte aqui detalhada é o reconhecimento feito pelo *software*; para ganhar velocidade no desenvolvimento, foi utilizado um banco de dados já populado fornecido pela iCoSys, Instituto de Sistemas Complexos. O sistema foi desenvolvido em Python, versão 3.8, por ser uma linguagem de fácil aprendizado e que fornece várias bibliotecas que facilitam o trabalho durante o desenvolvimento, e pode ser representado pelo fluxograma da Figura 11.

Figura 11 – Fluxograma do *software* total

Fluxograma do Projeto



Fonte: Acervo Pessoal (2020).

O evento se inicia a partir do momento em que um aparelho é ligado; o *hardware* adquire o sinal, um *software* trata e configura a assinatura, o sistema a recebe e faz a extração dos dados desta assinatura, ou seja, a partir das ondas geradas por esses aparelhos que estão ligados à rede, são extraídas informações, por meio de um processamento de sinal, utilizadas para a diferenciação de cada eletrodoméstico.

O dado adquirido passa pela *Random Forest*, algoritmo de classificação, que é encarregado de reconhecer o aparelho correspondente, por meio das informações coletadas. Caso o aparelho seja encontrado, com aproximadamente 90% de certeza, o usuário é informado sobre o eletrodoméstico. Se a taxa de precisão da resposta estiver entre 80% e 90%, a informação é

retornada para o usuário a fim de confirmar essa resposta.

Caso o dado não seja reconhecido, o usuário é questionado pelo sistema, sobre qual o aparelho foi ligado naquele momento, e os dados são salvos, juntamente com a informação do usuário no banco de dados.

5 RECONHECIMENTO E IDENTIFICAÇÃO DO ELETRODOMÉSTICO

Para a construção do código que faria reconhecimento do eletrodoméstico, primeiramente, buscamos um banco de dados já populado, com informações referentes a diversos aparelhos eletrônicos distintos (essa busca foi feita a fim de encontrar um banco de dados populado com dados reais para ser utilizado no desenvolvimento do trabalho).

O banco de dados utilizado foi o ACS - F1 da iCoSys, ele foi amostrado em uma frequência de 0,1 Hz e foi adquirido em duas sessões de 1 hora cada, que é um valor razoável de tempo para construir uma assinatura de consumo elétrico (IEEE, 2013).

As informações presentes no banco de dados utilizado seguem o padrão esperado na aquisição em tempo real dos aparelhos; o banco disponibiliza os dados em dois formatos diferentes, sendo eles: XML, que foi projetado para armazenar as observações brutas, alguns metadados e os valores verdadeiros das categorias de dispositivos, e em uma estrutura de dados MAT (IEEE, 2013).

O banco utilizado é composto por dez categorias de aparelhos distintos, são eles:

- a) geladeiras e Freezers;
- b) Tvs de tecnologia LCD;
- c) notebooks;
- d) computadores;
- e) sistemas de Hi-Fi com CD players;
- f) lâmpadas fluorescentes compactas (CFL);
- g) micro-ondas;
- h) máquinas de café;
- i) celulares via carregador de bateria;
- j) impressoras.

Dados esses aparelhos, foi adquirido o consumo dos seguintes parâmetros pela iCoSys:

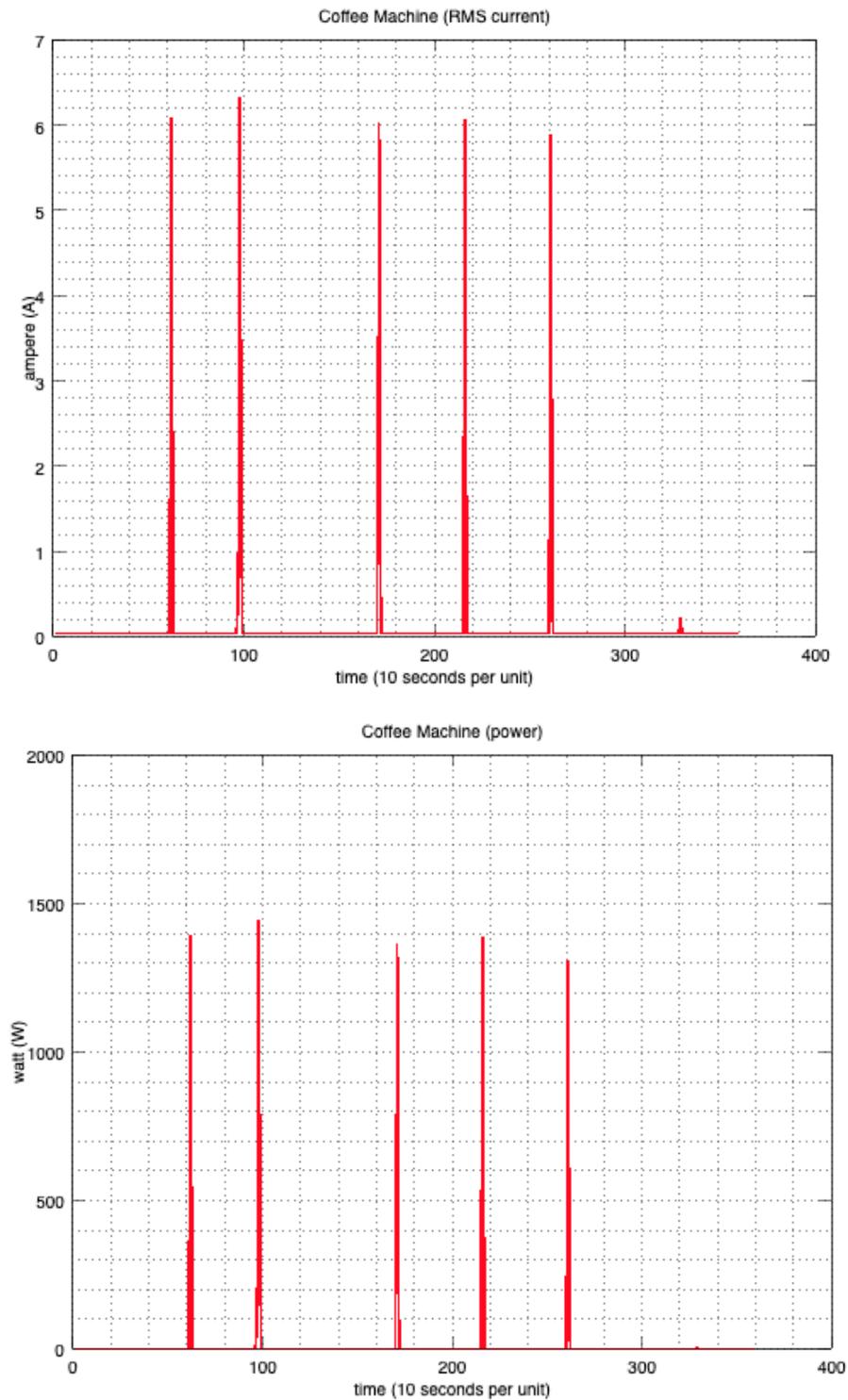
- a) potência real (W);
- b) potência reativa (Var);
- c) frequência (Hz);

- d) corrente RMS (A);
- e) tensão RMS (V);
- f) fase de tensão em relação à corrente.

As seis formas de onda que compõem o banco de dados são responsáveis pela assinatura da carga de cada um dos aparelhos, ou seja, o reconhecimento do eletrodoméstico se deu a partir do conjunto dos parâmetros citados anteriormente ou do tratamento de algum desses dados para a extração de outro mais relevante que pudesse compor a assinatura, como a potência máxima e mínima, ângulo da fase relacionado a corrente RMS, entre outros.

Assim, para melhor entendimento do assunto, pôde-se comparar as formas de onda dos aparelhos. Na Figura 12, pode-se observar um gráfico referente à corrente RMS de uma cafeteira em funcionamento e o gráfico da potência real respectivamente.

Figura 12 – Gráficos da corrente RMS e potência de uma cafeteira em funcionamento

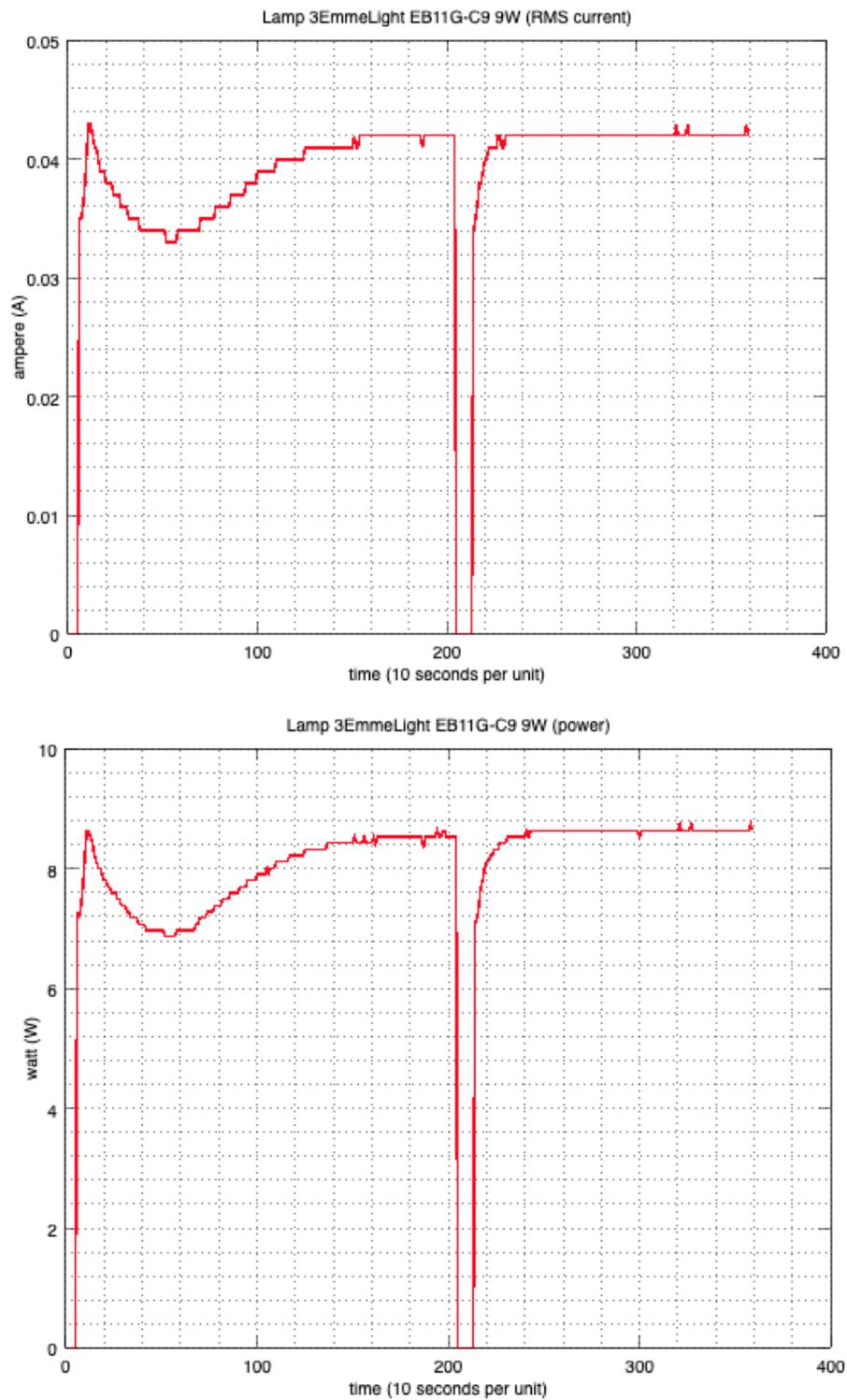


Fonte: IEEE (2013).

A seguir, observamos os sinais de uma lâmpada fluorescente. A Figura 13 mostra a corrente RMS e a potência real, respectivamente. É possível ver que os sinais da lâmpada e da cafeteira são distintos entre si. Foram apresentados apenas dois parâmetros, porém a assinatura

de uma carga é composta por uma série de parâmetros distintos dando mais singularidade ao aparelho.

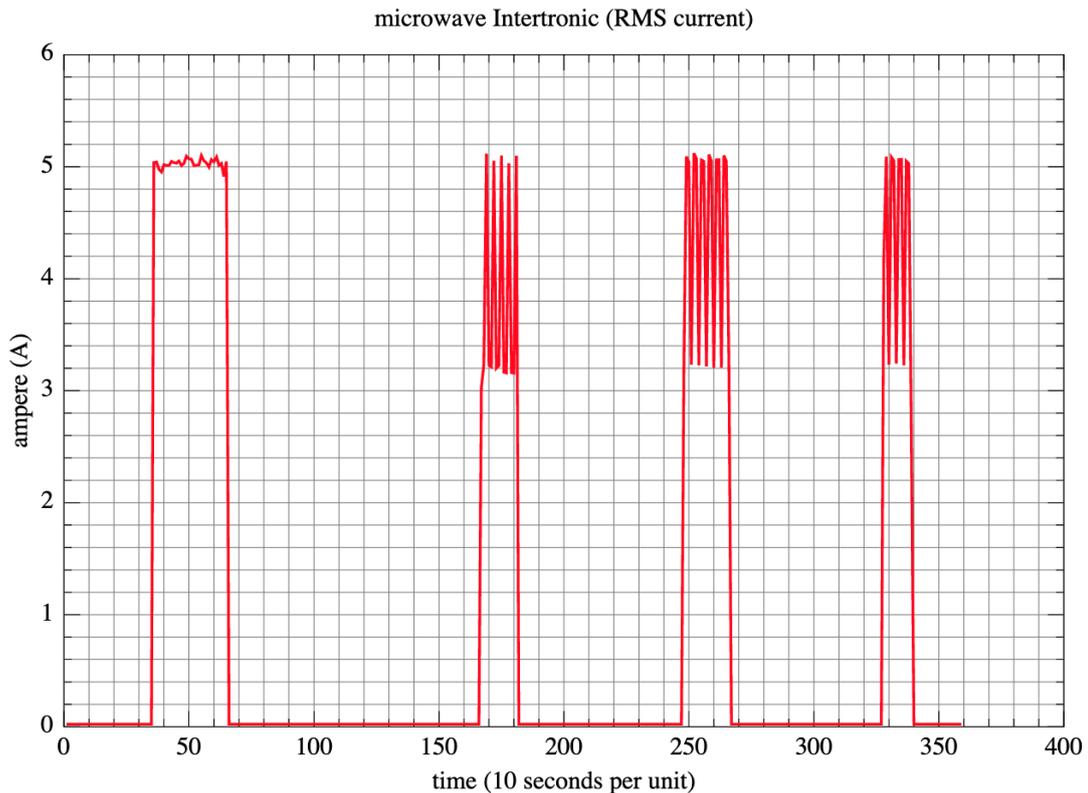
Figura 13 – Gráficos da corrente RMS e potência de uma lâmpada



Fonte: IEEE (2013).

Nessas análises de sinais, também é possível verificar alguns dados interessantes. Conforme observamos na Figura 14, o funcionamento de um micro-ondas é feito de ciclos, em que o pico inicial é o aparelho carregando; segundos após, o magnétron entra em ação. É a partir de análises como essas que conseguimos definir os dados mais relevantes para compor a assinatura da carga.

Figura 14 – Gráfico da corrente RMS de um micro-ondas em funcionamento



Fonte: IEEE (2013).

O código começou a ser construído com base nas formas de onda conhecidas e a partir da leitura do banco e de seus valores. A ação foi encontrar o diretório do banco de dados e varrer as pastas internas para leitura de cada dado.

Para uma melhor visualização de quais dados seriam mais importantes e quais decisões estariam presentes nos nós da árvore de decisão, optamos por escrever a média de cada grandeza por aparelho, referente à amostra do banco de dados, em um arquivo CSV (*Comma Separated Values*). Por fim, foram escritos os valores mínimos e máximos para cada uma das grandezas presentes no banco de dados para cada aparelho em outro arquivo CSV.

Com os dados organizados em planilhas, observamos que os valores brutos de corrente e tensão de cada aparelho não acrescentariam muito no reconhecimento do aparelho; já os dados extraídos de suas formas de onda agregaram conhecimento suficiente para reconhecê-lo.

Partindo deste princípio, foi estruturada a árvore de decisão. Primeiramente, foram definidas as decisões principais que iniciaram a árvore; com elas definidas, calculamos o ganho de informação para cada uma. Aquela que apresentar o maior ganho de informação será responsável pelo ramo inicial da árvore de decisão.

O ganho de informação é calculado a partir da entropia, que é um termo utilizado para caracterizar o grau de pureza; logo, a entropia de um conjunto de dados significa a pureza deles, ou falta de informação daquele conjunto. A entropia pode ser calculada com a Equação 14, (CAMPOS, 2017).

$$Entropia(S) = - \sum_{i=1}^n p_i * \log_2 p_i \quad (14)$$

Onde

entropia(S) = entropia do conjunto S

S = conjunto de treino

p = exemplos pertencentes ao conjunto

O ganho de informação utiliza da fórmula anterior em seu cálculo, o ganho é a redução esperada na entropia S. Sua fórmula pode ser visualizada na Equação 15.

$$Ganho(S, A) = Entropia(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} * Entropia(S_v) \quad (15)$$

O atributo que pertencerá ao topo da árvore de decisão é aquele que apresenta o maior valor para o ganho de informação.

Para a leitura dos dados do banco foi utilizado um código aberto feito em *Python* para o banco de dados da iCoSys, ele pode ser visualizado no Anexo A.

A partir desse código, foi possível extrair informações como os valores médio, mínimo e máximo de cada uma das grandezas para um arquivo CSV. Com os valores tabelados, foi possível visualizar melhor a influência de cada grandeza para a distinção do eletrodoméstico. O código utilizado para extrair essas informações pode ser observado no Apêndice A.

Com este código, podemos gerar uma tabela semelhante à Tabela 2, com os dados para cada eletrodoméstico presente no banco. A Tabela 2 é uma versão simplificada e minimizada da tabela original.

Tabela 2 – Dados médios

Eleto	Fase	Freq (Hz)	Pot Reativa (Var)	Pot Real (W)	Tensão (V)	Corrente (A)
microwave	215.17	50.0	22.79	199.62	234.69	0.97
comp-monitor	308.12	49.99	-24.85	39.14	227.92	0.21
hifi Cat RCD3839	49.61	50.0	4.17	4.06	224.11	0.02
printer HP 4650	334.84	50.0	-22.81	74.71	238.09	0.42
mobile Sony	4.10	50.0	-0.054	2.34	235.04	0.01
lamp	322.46	49.99	-3.83	7.86	224.76	0.03
fridge Bauknecht	23.98	50.00	81.66	67.58	225.20	0.48
coffeeMachine	268.58	50.00	-8.93	19.87	233.01	0.12
tv Panasonic	282.61	50.0	-50.86	18.30	236.80	0.24

Fonte: Acervo Pessoal (2020).

A partir da da Tabela 2, e da tabela de valores mínimos e máximos, foi possível chegar a decisões que iriam posteriormente compor a árvore de decisão.

Para fins de validação do método escolhido, foi elaborado um código visando calcular o ganho de informação de cada decisão - o código citado pode ser visualizado no Apêndice B.

Ao executar o código, obtivemos um resultado que desaprovava a teoria inicial; todos os parâmetros apresentaram um resultado muito semelhante para o ganho de informação, logo as "decisões" escolhidas inicialmente não eram relevantes o suficiente para a aplicação.

A fim de tentar novamente validar o método da árvore de decisão, foi utilizada a biblioteca Pandas, responsável por abrir e inspecionar uma variedade de conjunto de dados, e, também, a biblioteca *Sklearn* responsável pela aprendizagem de máquina. Foi desenvolvido um novo código que divide um conjunto de dados em recursos e variável de destino, depois divide o conjunto de dados em conjunto de treinamento e conjunto de teste; posteriormente, cria um objeto classificador da árvore de decisão e treina os dados; por último, prevê a resposta para um conjunto de dados em teste.

O código mostra a porcentagem de vezes que o algoritmo está correto em sua decisão, e pode ser visualizado no Apêndice C. Porém, o resultado utilizando esse método a partir dessas bibliotecas também não foi satisfatório; o resultado era baixo, com uma precisão em torno de 38%, o que não seria suficiente para distinguir os aparelhos, nem apresentaria relevância dentro do produto.

Após o insucesso com a árvore de decisão, optamos por testar o método *Radom Forest* (floresta aleatória), que é um método de aprendizagem de conjunto para uma classificação ou regressão. Consiste em uma infinidade de árvores de decisão no momento do treinamento (AYYADEVARA, 2018).

Ao iniciarmos a programação deste método, fizemos um retorno ao estudo buscando assimilar quais dados seriam úteis para tal aplicação. As variáveis escolhidas foram as mais básicas e superficiais para o primeiro teste do código, sendo elas: frequência média, potência média, potência reativa média, tensão média, corrente média e inclinação média do sinal. Para este primeiro teste, foram utilizados dez sinais distintos para a classificação.

Foram extraídos os dados, da mesma forma que foi apresentado anteriormente, a partir de um código para a leitura dos sinais, modelagem e escrita em um arquivo CSV. Após esses dados serem escritos no arquivo, executamos tal código para verificá-los. A Figura 15 também mostra a quantidade de linhas e colunas presente na tabela.

Figura 15 – Quantidade de linhas e colunas

```

1      # Usado para manipulacao de dados
2      import pandas as pd
3
4      # Le os dados e exhibe as primeiras 5 linhas
5      features = pd.read_csv('db_elet.csv')
6      features.head(5)
7      print('forma dos recursos:', features.shape)

```

Fonte: Acervo Pessoal (2020).

Para dar continuidade no código, foi preciso fazer outra modelagem dos dados já tratados. Colunas possuíam respostas no formato *string* foram modificadas para obedecer a uma codificação ativa, codificando os dados *one-hot* e utilizando a função *get dummies* da biblioteca Pandas. A biblioteca Pandas é responsável pela manipulação dos dados, contando com uma estrutura conhecida como *dataframe*, que é basicamente uma planilha de Excel com linhas e colunas (KOEHRSEN, 2017).

One-hot-encoding é a técnica de transformar os valores em variáveis, colunas e binárias (VAZ, 2018).

Para essa transformação foi utilizado o Código da Figura 16.

Figura 16 – Transformação de valores em variáveis

```

1      # Estatísticas descritivas para cada coluna
2      features.describe()
3
4      # Codifica os dados one-hot usando pandas get_dummies
5      features = pd.get_dummies(features)
6      # Exibe as primeiras 5 linhas das ltimas 12 colunas
7      features.iloc[:,5:].head(5)

```

Fonte: Acervo Pessoal (2020).

Após essa etapa, foi necessário separar os dados em recursos e destinos - recursos

são todas as colunas que são utilizadas no modelo para fazer a previsão e destinos são os dados que queremos prever. Existe também o valor que se espera prever, denominado alvo, rótulo, no caso é o eletrodoméstico. Para isso, convertamos os *dataframes* do Pandas em matrizes Numpy. na Figura 17 foi realizada a importação da biblioteca Numpy a fim de realizar a conversão para matrizes, remover os rótulos dos recursos e salvar os nomes dos recursos para utilização posteriormente.

Figura 17 – Importação da biblioteca e conversão para matrizes

```

1  # valores que queremos prever
2  labels = np.array(features['Label'])
3
4  # Remove os rotulos dos recursos
5  # eixo 1 refere-se s colunas
6  features= features.drop('Label', axis = 1)
7
8  # Salva nomes de recursos para usar posteriormente
9  feature_list = list(features.columns)
10
11 # Converter em numpy array
12 features = np.array(features)

```

Fonte: Acervo Pessoal (2020).

Na etapa final de preparação, os dados foram divididos em conjuntos de treinamento e teste. Os parâmetros *train features* e *train labels* formam o conjunto de dados utilizados para o treinamento do modelo, separamos 60% dos dados para serem utilizados no treinamento, já os parâmetros *test features* e *test labels* foram o conjunto de dados utilizados para o teste do modelo, representados por 40% dos dados, as porcentagens dos dados utilizados para o treinamento e para teste foram baseadas em outros códigos de *Random Forest* (FERNANDES; SILVA; OLESKOVICZ, 2010).

É necessário que o conjunto de teste represente um valor expressivo dentro do conjunto total de dados para que a precisão do método seja fiel à prática (VOLPI, 2019).

No treinamento, o comportamento ocorre semelhantemente a uma observação, onde o modelo visualiza as respostas de modo a aprender a partir dos recursos; posteriormente, é pedido para que o modelo faça previsões em um conjunto teste no qual ele só tem acesso aos recursos, e não as respostas. Em seguida, são comparadas as previsões que ele fez com as reais respostas para verificar a precisão do modelo treinado; podemos observar isso na Figura 18.

Figura 18 – Divisão em parâmetros de teste e de treinamento

```

1  # Divida os dados em conjuntos de treinamento e teste
2  train_features, test_features = train_test_split(features,
3                                              labels,
4                                              test_size = 0.6,
5                                              random_state = 60)
6  train_labels, test_labels = train_test_split(features,
7                                              labels,
8                                              test_size = 0.6,
9                                              random_state = 60)
10
11 # Instancia o modelo com 1000 rvores de decis o
12 rf = RandomForestRegressor(n_estimators = 2000, random_state = 42)
13 # Treine o modelo nos dados de treinamento
14 rf.fit(train_features, train_labels)
15
16 # Usa o m todo de previs o da floresta nas
17 predictions = rf.predict(test_features)
18 # Calcula os erros absolutos erros
19 errors = abs(predictions - test_labels)
20 # Imprime o erro m dio absoluto (mae)
21 print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.')
```

Fonte: Acervo Pessoal (2020).

Ao executar o código do Apêndice D, levando em conta dez sinais com as variáveis citadas anteriormente e mil árvores de decisão, obtivemos uma precisão de 63.7 % como conseguimos ver na Figura 19.

Figura 19 – Precisão obtida no primeiro teste do código *Random Forest*

```

1  The shape of our fetuares is: (11, 7)
2  Mean Absolute error: 2.28 degress.
3  [ 57.24285714  19.4875  32.175]
4  Accuracy: 63.7%
```

Fonte: Acervo Pessoal (2020).

Como a precisão ainda ficou abaixo do esperado, buscamos, em novas pesquisas, alguma forma de melhorá-la. Foi no artigo de Dyck *et al.* (2016) que pudemos verificar novos parâmetros para incluir no código, a fim de melhorar a precisão da resposta obtida pelo código.

Nesse modelo, foram incluídos dados como potência máxima e potência mínima do sinal do aparelho. Ao incluir essas duas novas colunas na extração dos dados a precisão aumentou para 80.28%, um aumento expressivo e já se responsabilizando por um resultado mais satisfatório que o primeiro, como pode ser visto na Figura 20.

Figura 20 – Precisão obtida no segundo teste do código *Random Forest*

```

1 The shape of our fetuares is: (10, 9)
2 Mean Absolute error: 1.5 degress.
3 [ 18.33333333  21.11111111]
4 Accuracy: 80.28%
```

Fonte: Acervo Pessoal (2020).

Buscando aumentar ainda mais a precisão e, como forma de teste, foi mudado o valor do número de árvores utilizadas na *Random Forest*, de mil para dois mil. O resultado foi positivo, porém ele não expressa muita relevância, passando de uma precisão de 80.28%, obtida anteriormente, para 80.35%, como pode ser visto na Figura 21.

Figura 21 – Precisão obtida no terceiro teste do código *Random Forest*

```

1 The shape of our fetuares is: (10, 9)
2 Mean Absolute error: 1.5 degress.
3 [ 17.9  21.4]
4 Accuracy: 80.35%
```

Fonte: Acervo Pessoal (2020).

O aumento do número de árvores utilizadas agregou também no tempo de processamento do código, quando testamos com mil árvores o tempo foi de aproximadamente 1.91 segundos, ao aumentar para duas mil árvores o tempo de processamento passou a ser aproximadamente 3.55 segundos, ocasionando um aumento de 85% em relação ao tempo anterior.

Acreditamos que para o produto final uma precisão em tronco de 80% não seria o ideal, seria preciso chegar em torno de 90% para que o resultado seja relevante. Hoyo-Montaño *et al.* (2018) define como sucesso aquele método que alcançar uma precisão acima de 90%.

Segundo Bacurau *et al.* (2014) uma boa precisão para esse tipo de aplicação é aquela que apresenta um erro menor que 10%, ou seja, onde a acurácia é representada por 90% ou mais.

Além dos estudos citados, Dyck *et al.* (2016) apresenta uma precisão de 95,07% utilizando da *Random Forest* para a classificação, o que mostra que o método pode trazer bons resultados.

Após obter resultados, com precisão satisfatória nos testes, a partir do banco de dados ACS-F1, optamos por realizar novos testes com um outro banco de dados, disponibilizado também pela iCoSys, chamado ASC-F2. O ASC-F2 utilizou uma frequência de amostragem de 0,1 Hz, e os dados foram adquiridos em duas sessões de uma hora, que é um tempo razoável

para a aquisição levando em conta a frequência de amostragem. As duas sessões de amostragem são realizadas para cada aparelho, que estão distribuídos por 15 categorias, são elas:

- a) geladeiras e freezers;
- b) sistemas Hi-Fi (com CD players);
- c) laptops;
- d) estações de computador (com monitores);
- e) lâmpadas fluorescentes compactas (CFL);
- f) micro-ondas;
- g) máquinas de café;
- h) telefones celulares (via carregador de bateria);
- i) impressoras;
- j) ventiladores;
- k) barbeadores;
- l) monitores;
- m) lâmpadas incandescentes;
- n) chaleiras.

O consumo dos aparelhos foram medidos em termos de:

- a) potência real (W);
- b) potência reativa (Var);
- c) frequência (Hz);
- d) corrente RMS (A);
- e) tensão RMS (V);
- f) fase de tensão em relação a corrente.

Ao fazer o teste com este outro banco de dados, obtivemos uma precisão de 80.2%, o que se mostrou mais uma vez, satisfatório. O resultado desse teste pode ser visualizado na Figura 22.

Figura 22 – Precisão obtida no quarto teste do código *Random Forest*

```
1 The shape of our fetuares is: (15, 9)
2 Mean Absolute error: 2.34 degress.
3 [ 3.88333333  11.91153846]
4 Accuracy: 80.2%
```

Fonte: Acervo Pessoal (2020).

6 CONSIDERAÇÕES FINAIS

Este trabalho de conclusão de curso visou estudar e validar formas para o reconhecimento de aparelhos ligados à rede; podemos dizer que o objetivo foi alcançado, uma vez que foi encontrado e aprovado o uso de um método para o reconhecimento dos eletrodomésticos, ainda que de cargas isoladas e fora do conjunto do produto final.

Com esta pesquisa, pudemos concluir que a utilização de uma quantidade maior de parâmetros para o desenvolvimento da assinatura de carga está atrelada em uma porcentagem maior na precisão final do reconhecimento do aparelho eletrônico. Pode-se concluir também que parâmetros como a frequência não é útil na análise de carga e a potência ativa trás ganhos significativos no estudo.

Foi possível notar que a primeira hipótese pensada e analisada para solucionar o problema do reconhecimento dos aparelhos foi rapidamente descartada por não apresentar resultados expressivos. Necessitamos de um conjunto de árvores de decisão para conseguirmos chegar a em uma solução mais precisa e expressiva para o produto.

Apesar da análise e reconhecimento dos aparelhos tratados neste trabalho ter muito para evoluir, antes de ser utilizada, de fato, no produto, podemos afirmar que o reconhecimento de um eletrodoméstico através de um conjunto de dados pelo método da árvore de decisão, atrelada a uma floresta aleatória, é um método possível e pode apresentar um resultado significativo.

O método da *Random Forest* pode significar um aumento no tempo de processamento, em relação ao método da árvore de decisão, de acordo com o número de árvores utilizadas no código. Esse é um fator importante para estudos posteriores, analisar o aumento do tempo de processamento e se o mesmo não interfere no funcionamento do produto.

Em trabalhos futuros, também, é interessante abordar o método utilizado inserindo mais dados para a comparação, como o tempo gasto em 20% do pico médio, potência de pico média, potência média 1 minuto antes do pico máximo, tempo gasto acima de X watts, inclinação do sinal antes e depois do pico máximo, entre outros fatores que irão incrementar ainda mais a assinatura de carga agregando maior precisão na hora do reconhecimento.

Futuramente é interessante adicionar na análise o transitório das cargas, analisando o período em que a carga é ligada e desligada, há a possibilidade de ganhar um aumento significativo na precisão final com essa análise.

Outro tópico interessante de abordar em trabalhos futuros é analisar as cargas através da *Random Forest* juntamente com um algoritmo de *Naive Bayes* para analisar se o resultado apresentado é satisfatório e se agrega algo no resultado final do *software*.

É muito interessante continuar a pesquisa para cargas acopladas, em funcionamento conjunto, em que é necessário distinguir um aparelho recebendo um sinal composto por mais aparelhos. Além disso, será importante pesquisar mais sobre a comunicação do *hardware* de

aquisição com o *software* de reconhecimento.

Para a conclusão do produto mencionado, é de grande interesse que pesquisas posteriores abordem a captação dos sinais através do *hardware* e que o método de reconhecimento seja, por fim, validado em cima desses sinais. A captação de sinais em trabalhos futuros deve ser desenvolvida através de aquisições com os *hardwares* mencionado, se aprofundando no estudo do processador proposto para ser utilizado. Os dados adquiridos deverão ser tratados através do processamento de sinal, a fim de deixá-lo pronto para o reconhecimento, livre de ruídos e interferências indesejadas.

REFERÊNCIAS

- ALCALM, A.; OLIVEIRA, C. A. d. S. **Fundamentos do Processamento de Sinais**. [S.l.]: Interciência, 2011. v. 1.
- ALTOÉ, L. *et al.* **Políticas públicas de incentivo à eficiência energética**. 2017.
- AYYADEVARA, V. K. **Random Forest**. In: *Pro Machine Learning Algorithms*. [S.l.]: Springer, 2018. p. 105–116.
- BACURAU, R. M. *et al.* **Medidor de energia inteligente para discriminação de consumo por aparelho através de assinatura de cargas**. 2014.
- BREIMAN, L. **Bagging predictors**. *Machine learning*, Springer, v. 24, n. 2, p. 123–140, 1996.
- BROWNLIE, J. **Naive Bayes for Machine Learning**. 2016.
- BROWNLIE, J. **Types of Classification Tasks in Machine Learning**. 2020.
- CAMPOS, R. **Árvores de Decisão**. 2017.
- CELESC. **Simulador de Consumo**. 2019.
- CHANG, H.-H.; LIN, C.-L.; YANG, H.-T. **Load recognition for different loads with the same real power and reactive power in a non-intrusive load-monitoring system**. In: IEEE. *2008 12th International Conference on Computer Supported Cooperative Work in Design*. [S.l.], 2008. p. 1122–1127.
- DUARTE, J. **O que é eficiência energética, como funciona e formas de economizar**. 2016.
- DYCK, C. *et al.* **Classifying Household Appliances From Electric Signals**. 2016.
- ECONOMY, A. for an E.-E. **Energy-Efficient Economy**. 2014.
- ENDALÉCIO, A. D. d. M. **Estudo e implementação de um sistema de levitação eletromagnética utilizando sensores de efeito Hall**. Universidade Federal do Rio de Janeiro, 2013.
- ENERGIA, B. **EPE reduz ganhos com eficiência energética em dez anos**. 2015.
- FERNANDES, R. A.; SILVA, I. N. d.; OLESKOVICZ, M. **Identificação de cargas lineares e não-lineares em sistemas elétricos residenciais usando técnicas para seleção de atributos e redes neurais artificiais**. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, SciELO Brasil, v. 21, n. 4, p. 389–405, 2010.
- FILHO, W. P. B.; AZEVEDO, A. d. **Impactos ambientais em usinas eólicas**. *Agrener GD*, v. 1, p. 1–17, 2013.
- GODOY, A. S. **ção à pesquisa qualitativa e suas possibilidades**. *RAE-revista de administração de empresas*, v. 35, n. 2, p. 57–63, 1995.
- HART, G. W. **Nonintrusive appliance load monitoring**. *Proceedings of the IEEE*, IEEE, v. 80, n. 12, p. 1870–1891, 1992.

HOYO-MONTAÑO, J. A. *et al.* **Non-intrusive electric load identification using wavelet transform.** *Ingeniería e Investigación*, 2018.

IEEE. **Appliance consumption signature database and recognition test protocols**, author=Gisler, Christophe and Ridi, Antonio and Zufferey, Damien and Abou Khaled, Omar and Hennebert, Jean, booktitle=2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA). In: . [S.l.], 2013. p. 336–341.

JORDAN, M. I.; MITCHELL, T. M. **Machine learning: Trends, perspectives, and prospects.** *Science*, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015.

JUNIOR, L. C. M.; COVOLAN, J. A. U. **APLICAÇÃO DE REDES NEURAIS PROFUNDAS PARA DETECÇÃO E CLASSIFICAÇÃO DE PLANTAS DANINHAS E SEU ESTADO DA ARTE.** 2018.

KOEHRSEN, W. **Random Forest in Python.** 2017.

LAKATOS, E. M.; MARCONI, M. de A. **Metodologia do trabalho científico: procedimentos básicos, pesquisa bibliográfica, projeto e relatório, publicações e trabalhos científicos.** [S.l.: s.n.], 2001. 43 p.

LUCIDCHART. **O que é um diagrama de árvore de decisão?** 2018.

MARDEGAN, C. **Capítulo I Transformadores de corrente, potencial e bobinas de Rogowski para fins de proteção–Parte.** *Revista O Setor Elétrico, São Paulo*, p. 22–31, 2010.

MARTINEZ, M. F. *et al.* **ção de consumo de energia elétrica através de conceitos greenbuilding.** *Revista Eletrônica de Potência*, v. 14, n. 2, p. 141–148, 2009.

NAKASHIMA, K. **Valor médio e eficaz.** *Setembro, 2013a. Universidade Federal de Itajubá*, 2007.

NASCIMENTO, F. A.; BERGER, P. A.; ROCHA, A. F. da. **Algoritmo para cálculo da distorção harmônica por meio da transformada rápida de hartley.** *a a*, v. 60, p. 4, 2007.

NASCIMENTO, F. P. do. **ção da Pesquisa. Natureza, método ou abordagem metodológica, objetivos e procedimentos.** *Brasília: Thesaurus*, 2016.

OLIVEIRA, A. C. de; LIMONGI, L.; CHAVES, D. **Áticas de Circuitos Elétricos.** 2010.

PAL, M. **Random forest classifier for remote sensing classification.** *International journal of remote sensing*, Taylor & Francis, v. 26, n. 1, p. 217–222, 2005.

PELLENZ, M. **Processamento Digital de Sinais (PDS).** 2005.

PUHLMANN, H. F. W. **Processamento Digital de Sinais.** 2020.

RANGEL-ABUNDIS, A. **Shunt and short circuit.** *Cirurgia y cirujanos*, v. 74, n. 1, p. 69–70, 2006.

RATTNER, H. **O esgotamento dos recursos naturais: catástrofe interdependência?** *Revista de Administração de Empresas, SciELO Brasil*, v. 17, n. 2, p. 15–21, 1977.

- ROBOCORE. **Sensor de Corrente ACS712 30A**. 2018.
- SENSE. **How Does Sense Detect My Devices?** 2016.
- SILBERSCHATZ, A. *et al.* **Database system concepts**. [S.l.]: McGraw-Hill New York, 1997. v. 5.
- SOUSA, T. C. *et al.* **ção de assinaturas de carga de eletrodomésticos residenciais em Smart Meters usando Inteligência Artificial e IoT: Implementação e testbed**.
- STOODI. **Fontes de Energia: tipos, fontes renováveis e não renováveis**. 2020.
- Texas Instruments. **EVM430-F6736 - MSP430F6736 EVM for Metering**. Texas, 2015.
- Texas Instruments. **MSP430™ ultra-low-power sensing measurement MCUs**. Texas, 2015.
- VAZ, A. L. **One-hot-encoding, o que é?** 2018.
- VERRI, L. **Administrando o consumo médio de energia em um lar padrão mundial classe média alta**. 2013.
- VIANNA, I. O. D. A. **Metodologia do Trabalho Científico**. São Paulo, p. 304, 2001.
- VOLPI, G. F. **Using Random Forest to Divide Data into Train and Test Assuring Representativeness**. *Novembro, 2019a. Medium*, 2019.
- YEGULALP, S. **What is Python? Powerful, intuitive programming**. 2019.

APÊNDICE A – CÓDIGO UTILIZADO PARA SALVAR OS DADOS EM ARQUIVO CSV

```
1 import xml.etree.ElementTree as ET
2 import os
3 import numpy as np
4 import csv
5
6 # retorna o diretorio de trabalho atual de um processo
7 directory = '/Users/brunamartini/code/power_signature' + '/ACS-F2'
8
9 # gera os nomes dos arquivos em uma arvore de diretorios
10 # percorrendo a arvore de cima para baixo ou de baixo para cima.
11 list_dir = [x[0] for x in os.walk(directory)]
12 list_dir = list_dir[1:]
13 list_name = []
14
15 # criar planilha e popular cabecalho
16 c = csv.writer(open("db_elet.csv", "w"))
17 c.writerow(["Label",
18            "phAngle_mean",
19            "Freq_mean",
20            "ReactPower_mean",
21            "Power_mean",
22            "Power_max",
23            "Power_min",
24            "Volts_mean",
25            "Cur_mean"])
26 a = 0;
27
28 for dirnames in list_dir:
29     list_filenames = [x[2] for x in os.walk(dirnames)][0]
30     for filenames in list_filenames:
31         # Se a extencao do agruivo for .mat
32         if filenames.endswith(".mat"):
33             cnt = 0
34             list_mat_phAngle = []
35             list_mat_freq = []
36             list_mat_power = []
37             list_mat_reacPower = []
38             list_mat_rmsCur = []
39             list_mat_rmsVolt = []
40             for line in open(dirnames + '/' + filenames):
41                 li=line.strip()
42                 if not li.startswith("#"):
43                     line_float = list(map(float, li.split(' ')))
```

```
44         cnt += 1
45         if cnt == 1:
46             list_mat_phAngle = line_float
47         elif cnt == 2:
48             list_mat_freq = line_float
49         elif cnt == 3:
50             list_mat_reacPower = line_float
51         elif cnt == 4:
52             list_mat_power = line_float
53         elif cnt == 5:
54             list_mat_rmsVolt = line_float
55         elif cnt == 6:
56             list_mat_rmsCur = line_float
57     if cnt != 6:
58         print(filenamees)
59
60     # salva valores medios em arquivo csv
61     a=a+1;
62     c.writerow([a,
63                 float(np.mean(list_mat_phAngle)),
64                 float(np.mean(list_mat_freq)),
65                 float(np.mean(list_mat_reacPower)),
66                 float(np.mean(list_mat_power)),
67                 float(np.max(list_mat_power)),
68                 float(np.min(list_mat_power)),
69                 float(np.mean(list_mat_rmsVolt)),
70                 float(np.mean(list_mat_rmsCur))])
```

APÊNDICE B – CÓDIGO UTILIZADO PARA O CÁLCULO DO GANHO DE DECISÃO

```

1 import xml.etree.ElementTree as ET
2 import os
3 import numpy as np
4 import csv
5 import math
6
7 # Ler arquivo CSV
8 # Calcular variaveis para arvore de decisao
9
10 def gain_of_information(variable, row_count, ent_p):
11     peso_1 = variable/row_count
12     peso_2 = (row_count-variable)/row_count
13
14     ent_f1 = -(math.log(1/variable))/math.log(variable)
15     ent_f2 = -(math.log(1/(row_count-variable)))/math.log((row_count-variable))
16
17     gain = ent_p - (peso_1*ent_f1 + peso_2*ent_f2)
18     return gain
19 class question:
20     def __init__(self, text, gain_of_information, position):
21         self.text = text
22         self.gain_of_information = gain_of_information
23         self.position = position
24
25 # calcula o numero de linhas de um arquivo CSV
26 # Subtrair 1 para retirar o cabecalho
27 row_count = sum(1 for line in open('db_elet.csv')) - 1
28 # Calcula entropia total da arvore
29 ent_p = -(math.log(1/row_count))/math.log(row_count)
30
31 frequencia = 0
32 angulo = 0
33 pot_r = 0
34 pot = 0
35 tensao = 0
36 corrente = 0
37
38 with open('db_elet.csv', 'r') as file:
39     reader = csv.reader(file)
40     for linha in reader:
41         # frequencia = 50
42         if float(linha[2]) == 50:
43             frequencia = frequencia + 1

```

```
44 # Angulo > 200
45 if float(linha[1]) > 200:
46     angulo = angulo + 1
47 # potencia reativa > 0
48 if float(linha[3]) > 0:
49     pot_r = pot_r + 1
50 # potencia > 50
51 if float(linha[4]) > 50:
52     pot = pot + 1
53 # tensao > 230
54 if float(linha[5]) > 230:
55     tensao = tensao + 1
56 # corrente > 0.1
57 if float(linha[6]) > 0.1:
58     corrente = corrente + 1
59
60 # ganho de informac o da frequencia
61 frequencia = print(gain_of_information(frequencia , row_count , ent_p))
62 angulo = print(gain_of_information(angulo , row_count , ent_p))
63 pot_r = print(gain_of_information(pot_r , row_count , ent_p))
64 tensao = print(gain_of_information(tensao , row_count , ent_p))
65 corrente = print(gain_of_information(corrente , row_count , ent_p))
```

APÊNDICE C – SEGUNDO TESTE DA PARA O MÉTODO DA ÁRVORE DE DECISÃO

```
1 # Load libraries
2 import pandas as pd
3 # Import Decision Tree
4 from sklearn.tree import DecisionTreeClassifier Classifier
5 # Import train_test_split function
6 from sklearn.model_selection import train_test_split
7 #Import scikit-learn metrics module for accuracy calculation
8 from sklearn import metrics
9 col_names = ["Eletro",
10             "phAngle_mean",
11             "Freq_mean",
12             "ReactPower_mean",
13             "Power_mean",
14             "Volts_mean",
15             "Cur_mean"]
16 # load dataset
17 pima = pd.read_csv("db_eletr.csv", header=None, names=col_names)
18 pima.head()
19
20 #split dataset in features and target variable
21 feature_cols = ["phAngle_mean",
22                "Freq_mean",
23                "ReactPower_mean",
24                "Power_mean",
25                "Volts_mean",
26                "Cur_mean"]
27 X = pima[feature_cols] # Features
28 y = pima.Eletro # Target variable
29
30 # Split dataset into training set and test set
31 # 70% training and 30% test
32 X_train = train_test_split(X, y, test_size=0.3, random_state=1)
33 X_test = train_test_split(X, y, test_size=0.3, random_state=1)
34 y_train = train_test_split(X, y, test_size=0.3, random_state=1)
35 y_test = train_test_split(X, y, test_size=0.3, random_state=1)
36
37 # Create Decision Tree classifier object
38 clf = DecisionTreeClassifier()
39
40 # Train Decision Tree Classifier
41 clf = clf.fit(X_train,y_train)
42
43 #Predict the response for test dataset
```

```
44 y_pred = clf.predict(X_test)
45
46 # Model Accuracy, how often is the classifier correct?
47 print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

APÊNDICE D – CÓDIGO UTILIZADO PARA O MÉTODO RANDOM FOREST

```
1 # Pandas usado para manipulacao de dados
2 import pandas as pd
3 # Use numpy para converter em matrizes
4 import numpy as np
5 # Usando Skicit-learn para dividir dados em conjuntos de treinamento e teste
6 from sklearn.model_selection import train_test_split
7 # Importe o modelo que estamos usando
8 from sklearn.ensemble import RandomForestRegressor
9 # Ferramentas de importacao necessarias para visualizacao
10 from sklearn.tree import export_graphviz
11 import pydot
12
13 # Leia os dados e exiba as primeiras 5 linhas
14 features = pd.read_csv('db_elet.csv')
15 features.head(5)
16 print('The shape of our features is:', features.shape)
17
18 # Estatisticas descritivas para cada coluna
19 features.describe()
20
21 # Codifique os dados one-hot usando pandas get_dummies
22 features = pd.get_dummies(features)
23
24 # Exibe as primeiras 5 linhas das ultimas 12 colunas
25 features.iloc[:,5:].head(5)
26
27 # Labels sao os valores que queremos prever
28 labels = np.array(features['Label'])
29
30 # Remova os rotulos dos recursos
31 # eixo 1 refere-se s colunas
32 features= features.drop('Label', axis = 1)
33
34 # Salvar nomes de recursos para usar posteriormente
35 feature_list = list(features.columns)
36
37 # Converter em numpy array
38 features = np.array(features)
39
40 # Divida os dados em conjuntos de treinamento e teste
41 train_features = train_test_split(features ,
42                                 labels ,
43                                 test_size = 0.2,
44                                 random_state = 60)
45 test_features = train_test_split(features ,
```

```
46         labels ,
47         test_size = 0.2,
48         random_state = 60)
49 train_labels = train_test_split(features ,
50                                 labels ,
51                                 test_size = 0.2,
52                                 random_state = 60)
53 test_labels = train_test_split(features ,
54                                 labels ,
55                                 test_size = 0.2,
56                                 random_state = 60)
57
58 print('Training Features Shape:', train_features.shape)
59 print('Training Labels Shape:', train_labels.shape)
60 print('Testing Features Shape:', test_features.shape)
61 print('Testing Labels Shape:', test_labels.shape)
62
63 # Instancie o modelo com 1000  rvores  de decisao
64 rf = RandomForestRegressor(n_estimators = 2000, random_state = 42)
65
66 # Treine o modelo nos dados de treinamento
67 rf.fit(train_features , train_labels)
68
69 # Use o metodo de previsao da floresta nas
70 predictions = rf.predict(test_features)
71
72 # Calcular os erros absolutos erros
73 errors = abs(predictions - test_labels)
74
75 # Imprime o erro medio absoluto (mae)
76 print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.')
77
78 # Calcular erro percentual absoluto medio (MAPE)
79 mape = 100 * (errors / test_labels)
80 print(mape)
81
82 # Calcular e exibir a precisao
83 accuracy = 100 - np.mean(mape)
84 print('Accuracy:', round(accuracy, 2), '%.')
```

ANEXO A – CÓDIGO UTILIZADO PARA EXTRAIR OS DADOS DO BANCO

```

1 #fonte: https://github.com/antonio86/ACS-F-readers/blob/master/README.md
2
3 import xml.etree.ElementTree as ET
4 import os
5
6 # retorna o diretorio de trabalho atual de um processo
7 directory = '/Users/brunamartini/code/power_signature' + '/ACS-F1'
8
9 # gera os nomes dos arquivos em uma arvore de diretorios
10 # percorrendo a arvore de cima para baixo ou de baixo para cima.
11 list_dir = [x[0] for x in os.walk(directory)]
12 list_dir = list_dir[1:]
13 list_name = []
14 for dirnames in list_dir:
15     list_filenames = [x[2] for x in os.walk(dirnames)][0]
16     for filenames in list_filenames:
17
18         # Se a extencao do arquivo for .xml
19         if filenames.endswith(".xml"):
20             list_name.append(filenames)
21             tree = ET.parse(dirnames + '/' + filenames)
22             root = tree.getroot()
23             list_xml_freq = []
24             list_xml_phAngle = []
25             list_xml_power = []
26             list_xml_reacPower = []
27             list_xml_rmsCur = []
28             list_xml_rmsVolt = []
29             for child in root:
30                 if child.tag == 'signalCurve':
31                     for child_2 in child:
32                         list_xml_freq.append(float(child_2.get('freq')))
33                         list_xml_phAngle.append(float(child_2.get('phAngle')))
34                         list_xml_power.append(float(child_2.get('power')))
35                         list_xml_reacPower.append(float(child_2.get('reacPower')))
36                         list_xml_rmsCur.append(float(child_2.get('rmsCur')))
37                         list_xml_rmsVolt.append(float(child_2.get('rmsVolt')))
38
39             check_xml = True
40             if (list_xml_freq != list_mat_freq):
41                 print(filenames)
42             if (list_xml_phAngle != list_mat_phAngle):
43                 print(filenames)
44             if (list_xml_power != list_mat_power):
45                 print(filenames)

```

```

46     if (list_xml_reacPower != list_mat_reacPower):
47         print(filenamees)
48     if (list_xml_rmsCur != list_mat_rmsCur):
49         print(filenamees)
50     if (list_xml_rmsVolt != list_mat_rmsVolt):
51         print(filenamees)
52     print(list_xml_phAngle)
53
54     # Se a extencao do arquivo for .mat
55     if filenamees.endswith(".mat"):
56         cnt = 0
57         list_mat_phAngle = []
58         list_mat_freq = []
59         list_mat_power = []
60         list_mat_reacPower = []
61         list_mat_rmsCur = []
62         list_mat_rmsVolt = []
63         for line in open(dirnames + '/' + filenamees):
64             li=line.strip()
65             if not li.startswith("#"):
66                 line_float = list(map(float, li.split(' ')))
67                 cnt += 1
68                 if cnt == 1:
69                     list_mat_phAngle = line_float
70                 elif cnt == 2:
71                     list_mat_freq = line_float
72                 elif cnt == 3:
73                     list_mat_reacPower = line_float
74                 elif cnt == 4:
75                     list_mat_power = line_float
76                 elif cnt == 5:
77                     list_mat_rmsVolt = line_float
78                 elif cnt == 6:
79                     list_mat_rmsCur = line_float
80
81             if cnt != 6:
82                 print(filenamees)
83
84             print('ok')
85             if (filenamees[-5] != child.get('session')):
86                 child.set('session', filenamees[-5])
87                 tree.write(dirnames + '/' + filenamees)
88
89             print(list_mat_phAngle)
90
91 for i in range(0,len(list_db)):
92     print(list_db[i] + '\t' + list_name[i])

```