

# IESTI01 – TinyML - 2021.2

## Relatório do Projeto Final da Disciplina

### 1. Título do Projeto

Identificação de cargas elétricas pelos sinais de corrente e tensão utilizando técnicas de Machine Learning

### 2. Nome dos Integrantes e Instituição

Carlos Eduardo Gomes - IFSP

Carlos Eduardo Oliveira Silva - IFSP

Ivan Lucas Arantes - IFSP

Matheus Mascarenhas - IFSP

### 3. Objetivo

Reconhecimento do tipo de aparelho doméstico em utilização na instalação elétrica residencial, através da coleta de valores instantâneos de corrente e tensão (assinatura de corrente e tensão).

### 4. Introdução

A demanda por energia elétrica cresce em ritmo acelerado motivado pelo crescimento populacional. A utilização de combustíveis fósseis na geração de energia elétrica não é sustentável. Comitês internacionais, como o IEC - International Electrotechnical Commission declara que o uso inteligente e economicamente viável de eletricidade, como fonte primária, é o mais importante fator para a solução do problema energético.

Diversas pesquisas vêm sendo conduzidas para a utilização eficiente da energia nos segmentos industriais, comerciais e residenciais. No segmento industrial foca-se em sistemas de gerenciamento energético, na comercial em sistemas de monitoramento do consumo energético das instalações (Prédios, centros comerciais, dentre outros). Para o setor residencial um aumento na eficiência é alcançado através da mudança comportamental dos consumidores, tarefa que é efetuada analisando-se a utilização da energia elétrica.

Considerando-se ainda, que diferentes equipamentos/eletrodomésticos geram assinaturas de potência, corrente e tensão distintas, torna-se possível a sua classificação pela sua utilização.

O presente projeto propõe o reconhecimento de assinaturas de diferentes dispositivos utilizando as variação das próprias amostras dos sinais de tensão e corrente (dados RAW), além da análise espectral. Todo o tratamento dos dados e o processo de inferência através de técnicas de Machine Learning baseadas em rede neural foi realizado em um sistema embarcado de baixa potência, no caso a placa ESP-32.

## 5. Descrição do projeto

O projeto consiste na identificação de cargas elétricas, através da aquisição de dados de tensão e corrente consumidos por ela. Pode-se dividir o desenvolvimento do projeto em um fluxograma mais simplificado de quatro etapas conforme apresentado na figura 1:



Fig 1 – Fluxograma de desenvolvimento do modelo embarcado

Para o projeto em questão criou-se a própria base de dados (dataset), portanto fez-se um algoritmo para coleta dos dados e envio para porta de comunicação serial. Os dados gerados não foram aplicados em tempo real diretamente no Edge Impulse via serial, mas foram criados arquivos em formato “.csv” que posteriormente eram carregados. Após coleta de dados, treinamento e pré-testes no próprio Edge Impulse utilizando as amostras de testes, o modelo pode então ser carregado no sistema embarcado (ESP-32) para os testes de inferência. Para uma boa adequação dos resultados finais, todo este fluxograma foi repetido inúmeras vezes.

### a. Diagrama de blocos

A seguir é mostrado o diagrama de blocos do sistema para a realização da inferência:

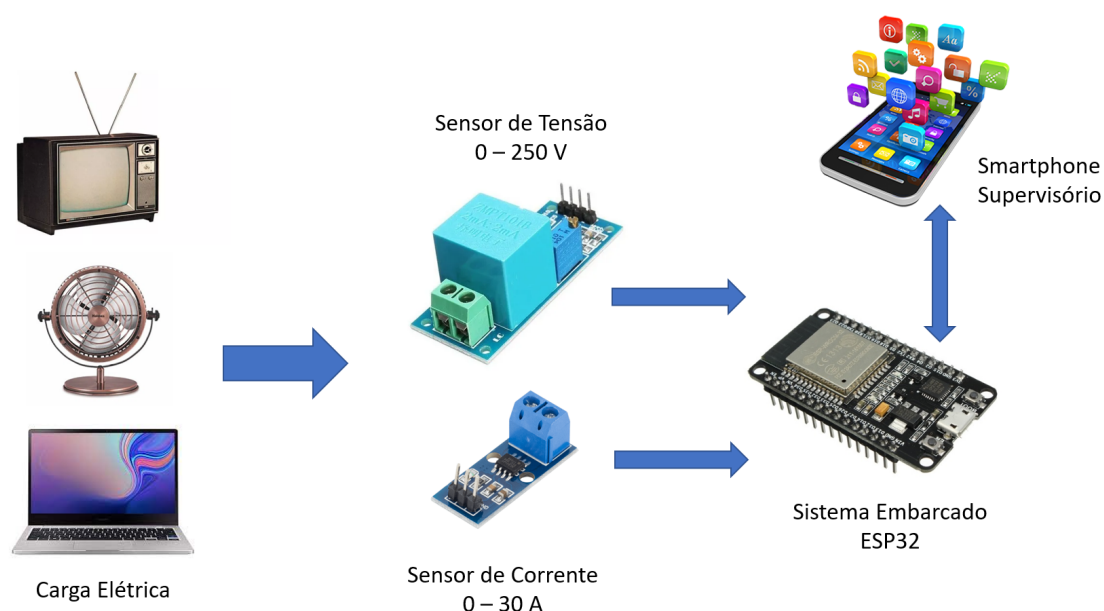


Fig 2 - Diagrama de blocos do hardware

## b. Hardware

O hardware desenvolvido, conforme figura 2, tem os seguintes componentes principais:

b.1- Sistema embarcado: Placa ESP-32 com microcontrolador que realiza a aquisição dos dados, tratamento e faz a inferência, de forma a identificar a carga elétrica ligada ao sistema. Uma interessante característica do ESP-32 utilizada no projeto é sua capacidade de comunicação Wifi

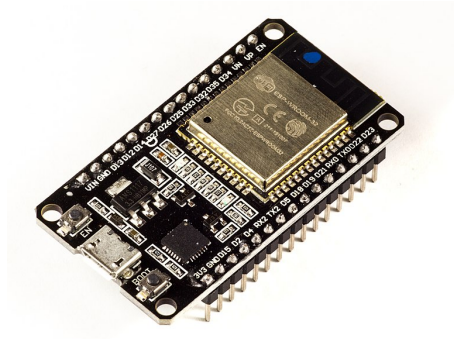


Fig 3 - Placa ESP32

(fonte: [https://commons.wikimedia.org/wiki/File:ESP32\\_Espressif\\_ESP-WROOM-32\\_Dev\\_Board.jpg](https://commons.wikimedia.org/wiki/File:ESP32_Espressif_ESP-WROOM-32_Dev_Board.jpg))

O diagrama de blocos do ESP-32 pode ser visto a seguir:

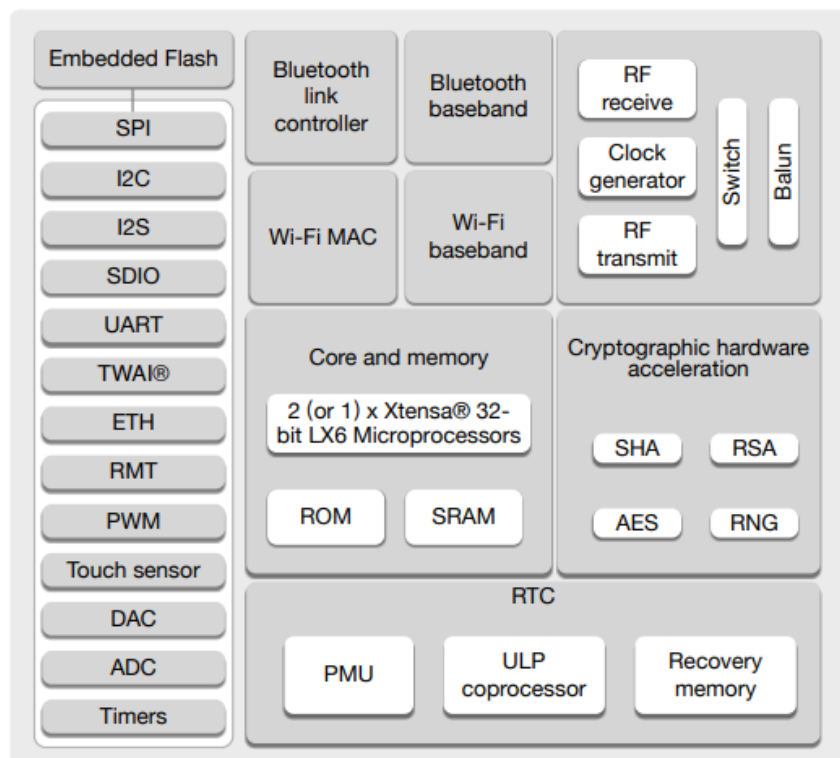


Fig 4 - ESP32

(fonte: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf))

b.2- Sensor de corrente: Modelo ACS712 (30A), para fazer a aquisição do sinal de corrente. Este módulo sensor de corrente já possui um off-set de 2.5Vdc, facilitando a leitura de correntes alternadas (que possuem semiciclo negativo).

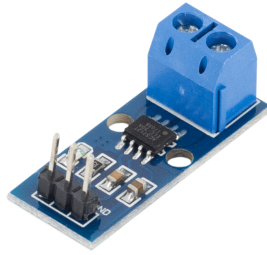


Fig 5 - Módulo sensor de corrente

(fonte: <https://www.filipeflop.com/wp-content/uploads/2017/07/001-3.png>)

b.3- Sensor de tensão: Módulo Sensor Tensão Ac 0 a 250 V Zmpt101b, para aquisição do sinal de tensão. Também possui a característica de off-set de 2.5Vdc.

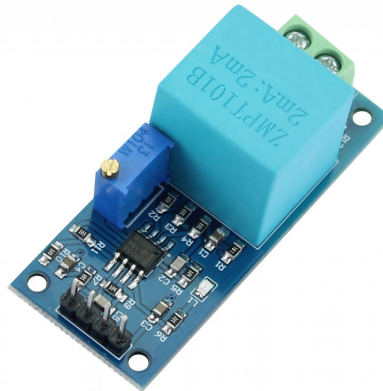


Fig 6 - Módulo sensor de tensão

(fonte: [https://www.usinainfo.com.br/1019779-thickbox\\_default/sensor-de-tensao-ac-zmpt101b-voltmetro-arduino.jpg](https://www.usinainfo.com.br/1019779-thickbox_default/sensor-de-tensao-ac-zmpt101b-voltmetro-arduino.jpg))

b.4- Smartphone: para realizar a interface com o usuário e apresentar a inferência realizada pelo modelo utilizou-se um smartphone com sistema operacional Android.

### c. Data collection

Conforme mencionado, a coleta das amostras de tensão e corrente foi efetuada através da placa ESP-32 executando um algoritmo desenvolvido na IDE do Arduino, especificamente para o projeto. Os dados gerados foram carregados no Edge Impulse por meio de arquivos “.csv”. Foram adquiridos sinais de 3 aparelhos (ventilador, notebook e televisão) além da aquisição a vazio (sem nenhuma carga conectada). Para cada classe foram adquiridos 2 minutos de amostras.

#### d. Preprocessing

Os dados registrados foram tensão (V), corrente elétrica (A) e potência instantânea (Pi - que foi obtida pelo produto de V e A no próprio programa de coleta de dados). Com uma taxa de aquisição de 1Khz, foi registrado uma mostra de cada sinal no intervalo de 1ms, resultando em 120.000 amostras por classe. Lembrando que o sinal elétrico da rede possui uma frequência de 60Hz. A tabela mostra um exemplo dos dados gravados em arquivo csv:

timestamp	V	A	Pi
0	-144.68	-1.21	175.62
1	-128.67	-1.08	139.03
2	-89.33	-1.06	94.82
3	-51.73	-0.41	21.41
4	-8.91	-0.51	4.54
5	57.24	-0.11	-6.24
6	104.23	0.27	28.34
7	143.57	0.48	69.11
8	154.02	0.46	71.21
9	138.7	0.73	101.11
10	89.26	0.88	78.68

Tab1 - Exemplo dos sinais amostrados de um ventilador

Um fato importante é que na coluna timestamp é mostrado o valor real medido em (ms) do tempo de aquisição das amostras, isto foi necessário para realmente se garantir que o período de amostragem estava correto.

A seguir são mostrados os gráficos de tensão e corrente de cada aparelho.

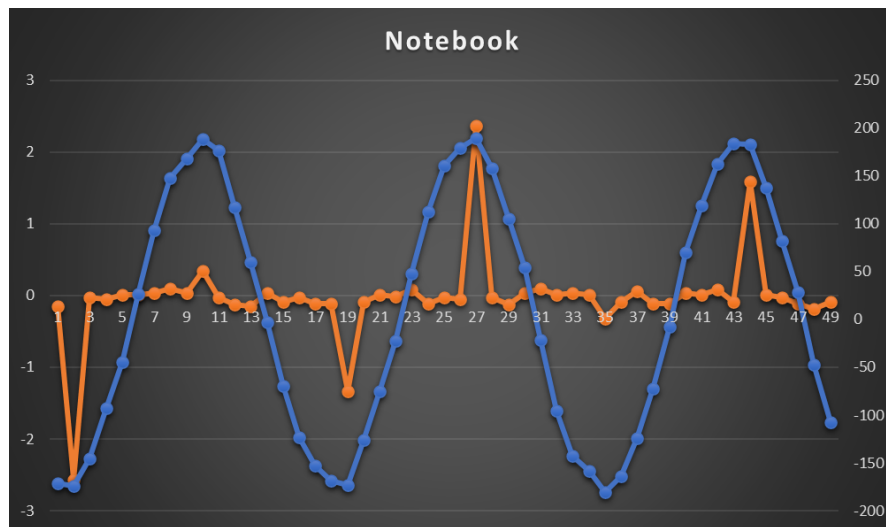


Fig 7 - Sinais de tensão e corrente de um notebook

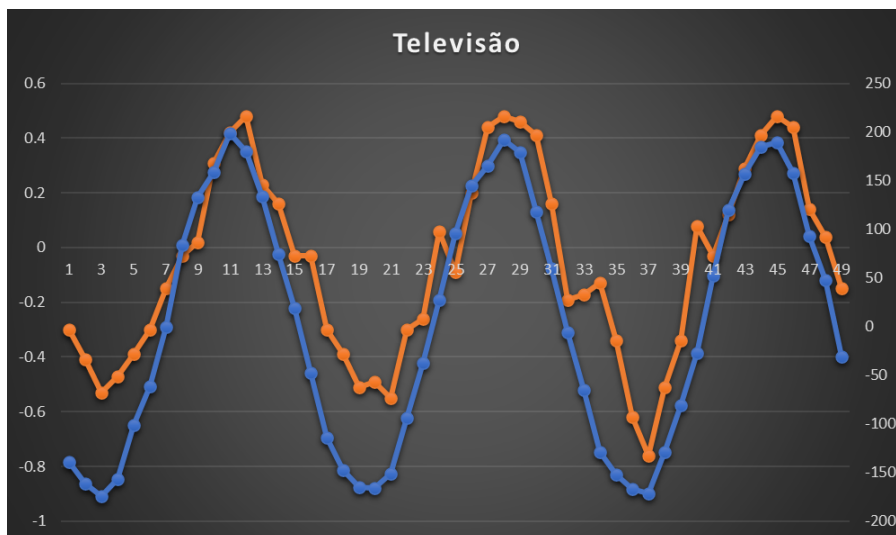


Fig 8 - Sinais de tensão e corrente de uma televisão

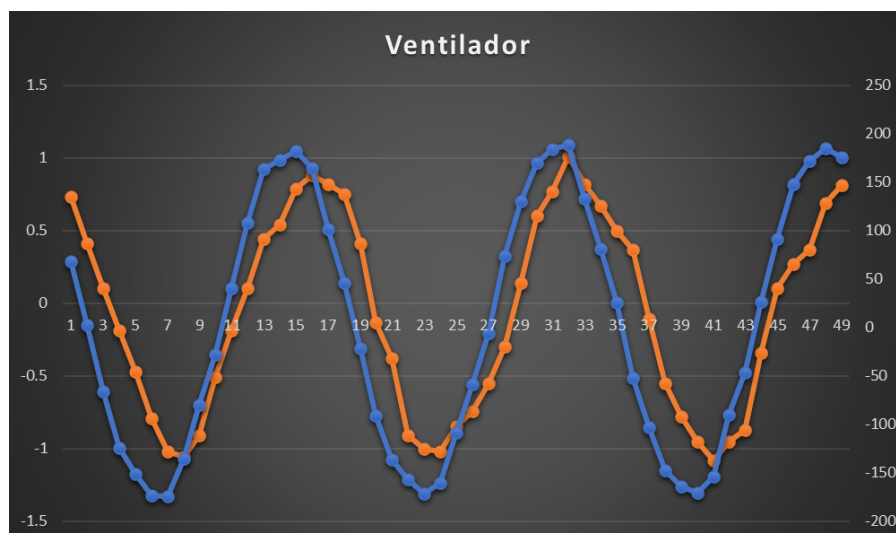


Fig 9 - Sinais de tensão e corrente de um ventilador

Os dados carregados já estão em termos de valores reais de tensão e corrente, isto foi feito no programa de coleta, subtraindo os valores de offset de 2.5v e multiplicando pelo fator de correlação (V/A) e (V/V) respectivamente dos módulos sensores.

Com relação ao processamento pelo Edge Impulse decidiu-se não usar o sinal de tensão, pois o mesmo é originário da rede elétrica e não é correlacionada a carga. Foram aplicadas janelas de 200ms com incremento de 20ms, e como entrada para a rede neural o resultado da análise espectral da corrente e potência instantânea, além das amostras propriamente ditas (RAW). As figuras mostram as configurações no Edge Impulse:

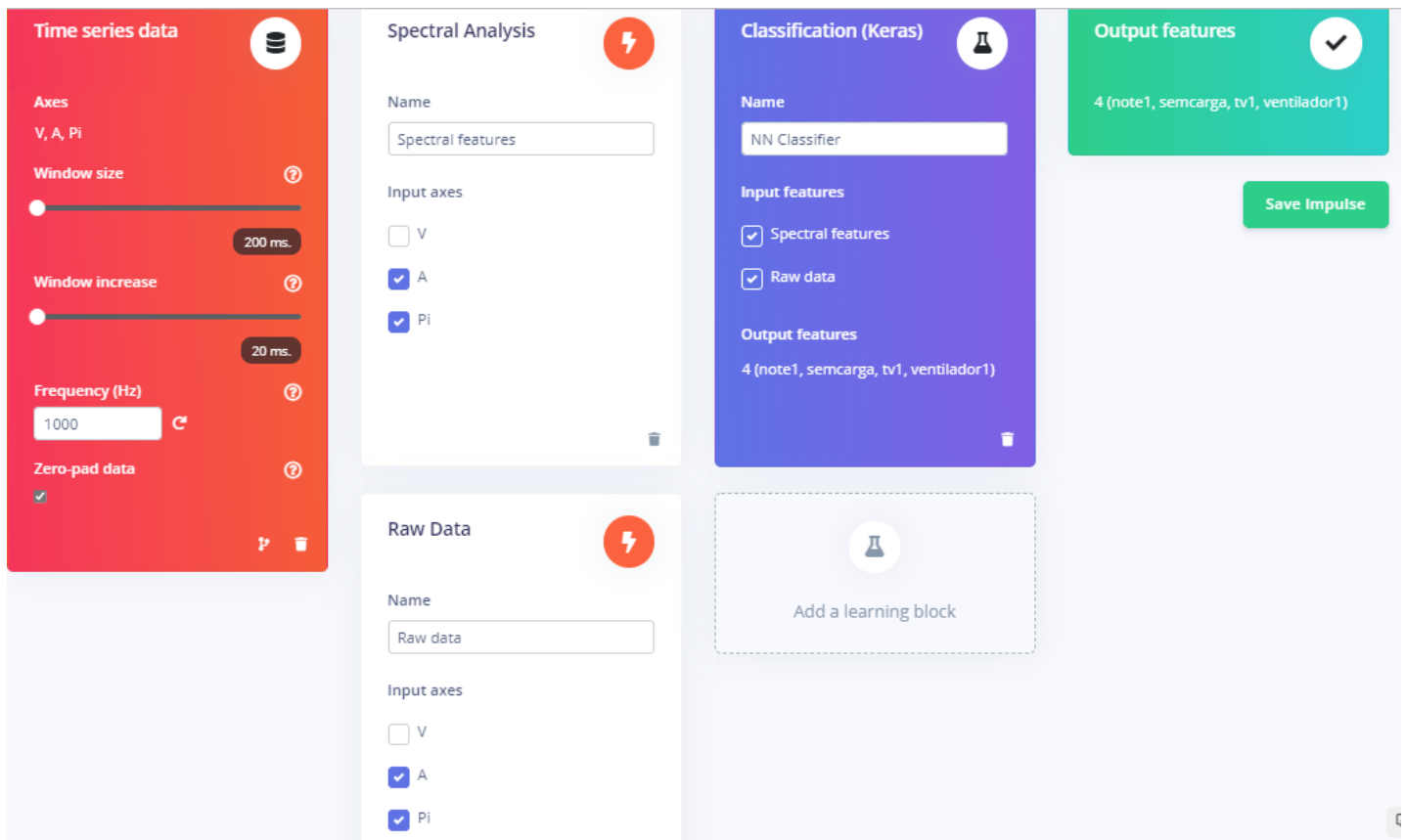


Fig 10 - Configurações de processamento no Edge Impulse

A seguir figuras da distribuição dos dados.

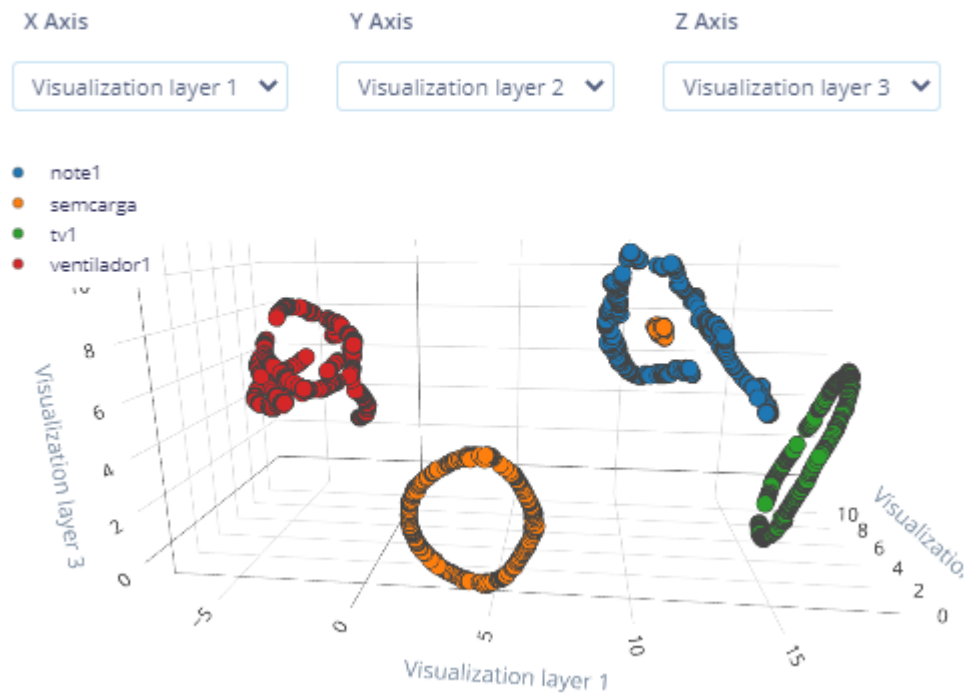


Fig 11 - Distribuição dos dados (RAW)





Fig 12 - Distribuição pela análise espectral

#### e. Model design

Tomando como base 200ms de janela e 1ms por amostra, então tem-se 200 “features” das amostras de corrente e 200 de potência instantânea, somado a 11 de cada uma como resultado da análise espectral, totalizando:  $200+200+11+11= 422$ . Estas 422 “features” são as entradas para a rede neural. Resolveu-se utilizar apenas uma camada oculta com os mesmos 422 neurônios. Após foi aplicado um Dropout com taxa de 0,1 a fim de evitar um overfitting. Por fim, a camada de saída trata-se de um classificador com 4 classes de saídas, que são os 4 labels definidos no projeto.

Neural Network settings

Training settings

Number of training cycles ?

50

Learning rate ?

0.0005

Neural network architecture

Input layer (422 features)

Dense layer (422 neurons)

Dropout (rate 0.1)

Add an extra layer

Output layer (4 classes)

Fig 13 - Configuração da rede neural

O resultado do treinamento é visto na figura 14:

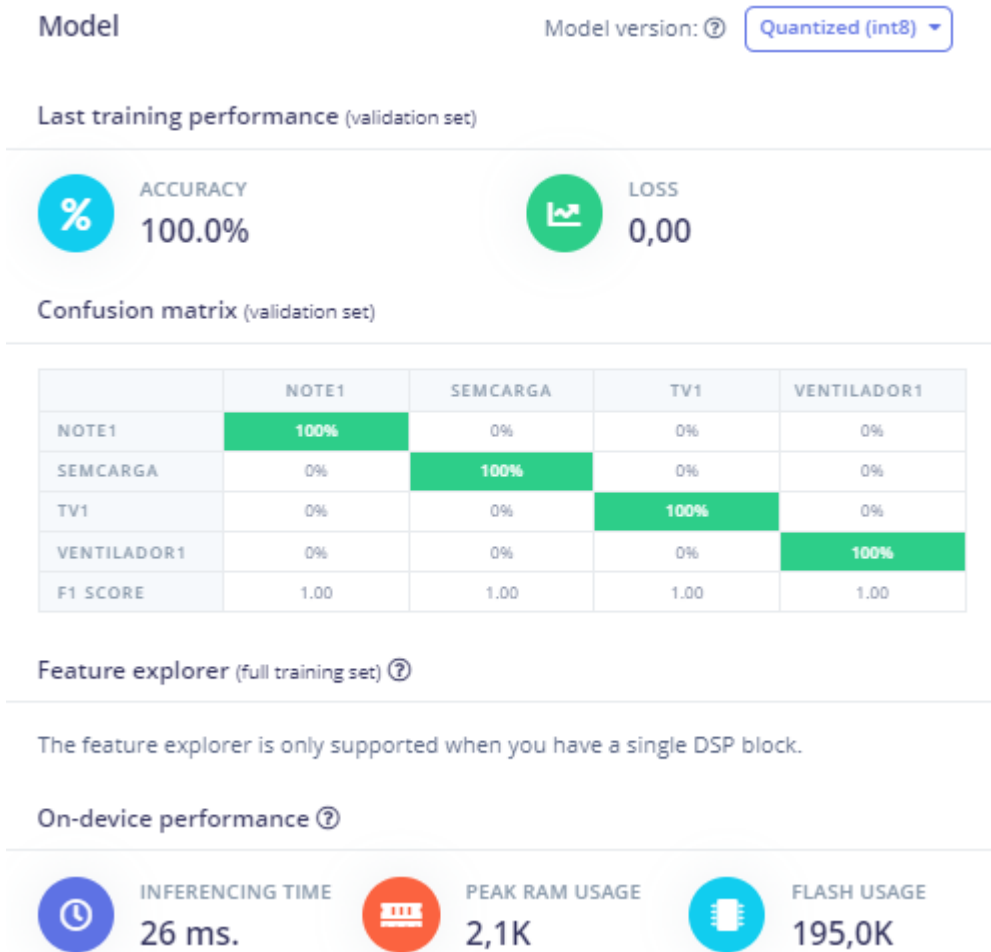


Fig 14 - Resultado do treinamento da rede

A seguir é visto o resultado de testes do modelo com dados de 20 segundos.

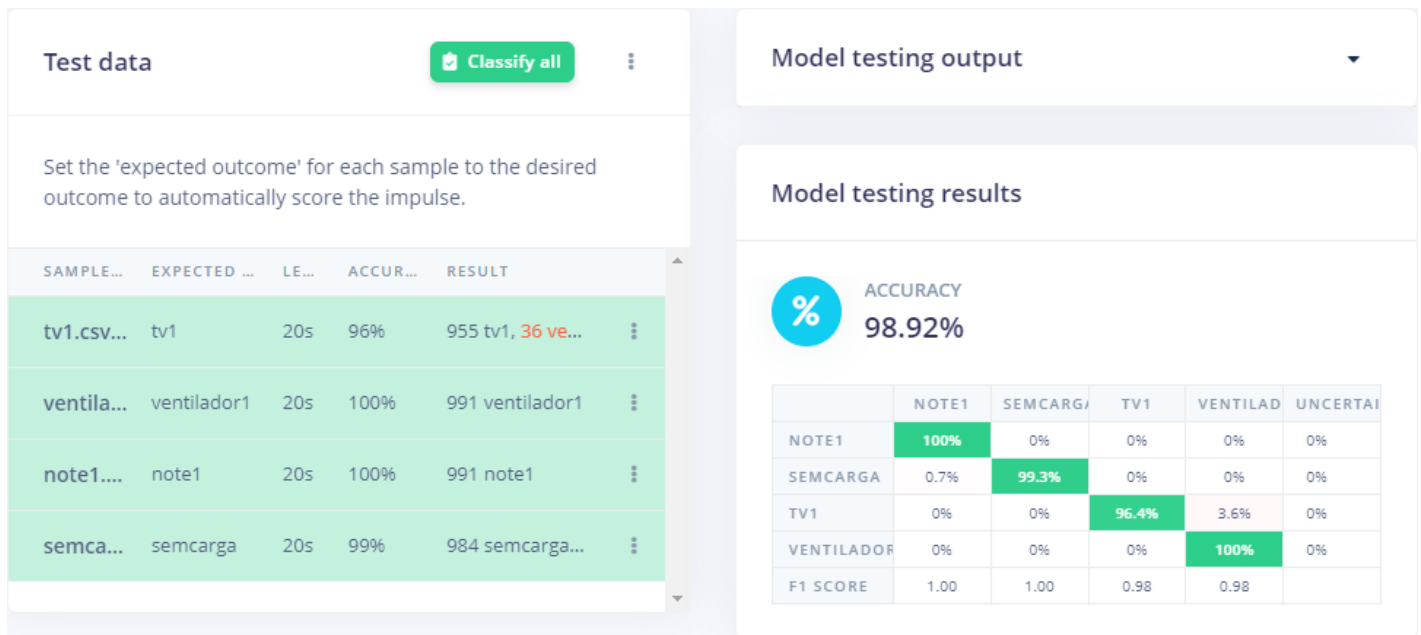


Fig 15 - Resultado de testes do modelo

Link para acesso ao projeto no Edge Impulse: <https://studio.edgeimpulse.com/studio/67114>

#### f. In system inference (Deploy)

Após o desenvolvimento e validação em computador, o modelo pode então ser convertido usando a ferramenta Edge Impulse. A conversão transforma o modelo em uma forma reduzida e a biblioteca para Arduino pode ser carregada na placa ESP-32. Foi então desenvolvido um algoritmo que integrou o processamento da inferência juntamente com a parte de comunicação wi-fi, de forma que o resultado fosse transmitido. Para a visualização da classificação fez-se um aplicativo utilizando a ferramenta Kodular (<https://www.kodular.io/>) assim o celular é capaz de receber e mostrar o resultado em tempo real. A tela do aplicativo é vista a seguir:

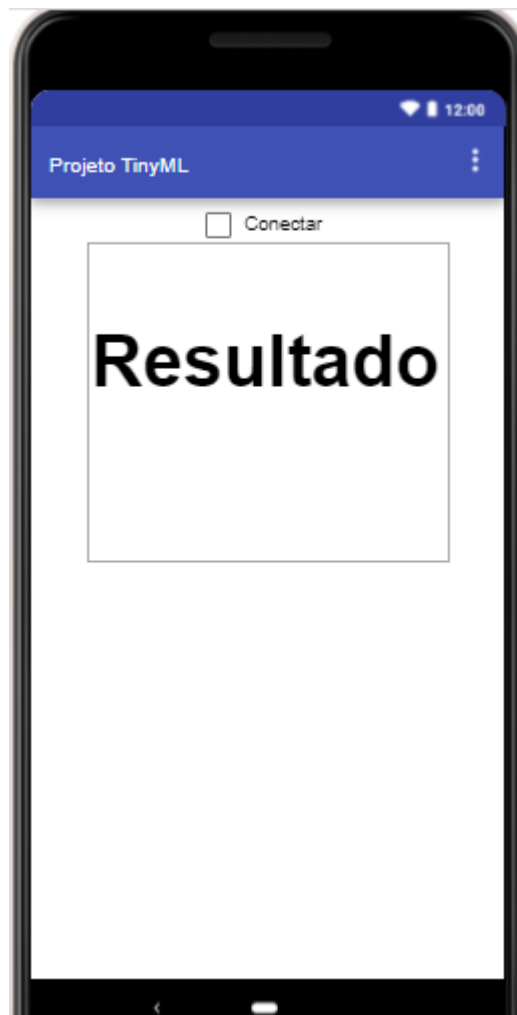


Fig 16 - Aplicativo para mostrar o resultado da inferência

Um vídeo demonstrativo de um processo de classificação das cargas pode ser observado neste link:

(<https://youtu.be/0CfNwiZa0Xg>)

## 6. Problemas solucionados:

Pode-se destacar dois problemas principais que foram analisados e solucionados:

- a) Frequência de amostragem: Inicialmente para a coleta de dados estava-se utilizando uma velocidade de comunicação serial de 115200 bits/seg, mas isto ocasionava um período de amostragem um pouco maior do que 1ms. Isto gerou uma baixa “accuracy” no treinamento. Para resolver esse problema aumentou-se a taxa para 500 kbits/seg e, após análise do tempo (timestamp), verificou-se que o intervalo de 1ms por amostra estava garantido.
- b) Sinal de tensão: Conforme mencionado a utilização dos sinais de tensão e corrente não se mostrou satisfatória, portanto acrescentou-se como entrada do modelo a potência instantânea. Posteriormente retirou-se a informação da tensão, melhorando um pouco a performance do modelo.

## 7. Conclusões:

Este trabalho apresentou o projeto e os resultados de um sistema de identificação de cargas elétricas utilizando os sinais elétricos de tensão e corrente. Para tanto fez-se uso de Machine Learning embarcado em uma placa ESP-32. Os resultados demonstram a validade da proposta, que pode ser empregada em pesquisas para a utilização eficiente da energia podendo contribuir como parte da solução para o problema energético. O sistema se mostrou válido e promissor, mas ainda precisa ser trabalhado para uma menor susceptibilidade às variações da rede elétrica, além da potência da carga, onde por exemplo o notebook pode sofrer uma variação de acordo com o ciclo de carga da bateria. Como trabalho futuro propõe-se o desenvolvimento de um sistema capaz de detectar mais de uma carga ligada ao mesmo tempo.

## 8. Referências

Kim, Jihyun, Thi-Thu-Huong Le, and Howon Kim. "Nonintrusive Load Monitoring Based on Advanced Deep Learning and Novel Signature." *Computational Intelligence and Neuroscience* 2017 (2017): 4216281-22. Web.

Liu, Yanchi, Xue Wang, and Wei You. "Non-Intrusive Load Monitoring by Voltage-Current Trajectory Enabled Transfer Learning." *IEEE Transactions on Smart Grid* 10.5 (2019): 5609-619. Web.

Seera, Manjeevan, Chee Peng Lim, Saeid Nahavandi, and Chu Kiong Loo. "Condition Monitoring of Induction Motors: A Review and an Application of an Ensemble of Hybrid Intelligent Models." *Expert Systems with Applications* 41.10 (2014): 4891-903. Web.