



# Modelo de Consumo de Energia baseado em Aprendizado de Máquina para Interfaces de Redes Celulares

*Leonardo N. Nascimento      Gabriel S. Enoki  
Carlos A. Astudillo*

Relatório Técnico - IC-PFG-24-40  
Projeto Final de Graduação  
2024 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.  
O conteúdo deste relatório é de única responsabilidade dos autores.

# Modelo de Consumo de Energia baseado em Aprendizado de Máquina para a Interfaces de Rede Celulares

Leonardo N. Nascimento      Gabriel S. Enoki      Carlos A. Astudillo

## Resumo

O relatório apresenta um modelo de consumo energético baseado em aprendizado de máquina voltado para dispositivos de redes celulares. Neste estudo, foram implementadas duas abordagens. A primeira concentra-se na previsão dos picos de consumo de potência, simplificando o problema ao focar nos momentos de maior demanda energética, enquanto a segunda considera o consumo de potência como função do tempo, permitindo previsões mais detalhadas e capturando padrões dinâmicos. O processo incluiu etapas de pré-processamento dos dados, como aplicação de filtros passa-baixa para suavizar sinais, remoção de anomalias com transformada wavelet e organização dos dados para treinamento em redes neurais. Os modelos utilizam parâmetros como largura de banda, potência de transmissão e recepção para prever o consumo energético. Os resultados demonstraram alto desempenho dos modelos, com coeficiente de determinação R-squared acima de 0,94 e MAE (*mean absolute error*) abaixo de 0,0135 para ambos os modelos. O segundo modelo destacou-se por sua capacidade de prever com maior precisão a forma da curva e o consumo máximo. Conclui-se que essas abordagens podem contribuir significativamente para a avaliação de desempenho e eficiência energética em dispositivos celulares. Esses modelos podem promover maior sustentabilidade e eficiência nas redes celulares, onde a maioria dos dispositivos opera com baterias com capacidade limitada.

## 1 Introdução

A evolução das redes de comunicação sem fio tem impulsionado o crescimento de aplicações inovadoras e sustentado o aumento exponencial de dispositivos conectados, particularmente na Internet das Coisas (IoT, Internet of Things). Nesse contexto, as redes celulares 5G e, futuramente, 6G, têm desempenhado um papel fundamental, oferecendo maior capacidade de transmissão, latência reduzida e suporte a um número significativo de dispositivos conectados simultaneamente. Entretanto, com essas inovações, surgem desafios importantes, sendo o gerenciamento de recursos de rádio (RRM, Radio Resource Management) um dos mais críticos. O RRM é responsável por coordenar o acesso aos recursos limitados da rede, garantindo níveis aceitáveis de qualidade de serviço (QoS, Quality of Service) mesmo em cenários de alta demanda, como aqueles apresentados pelo acesso massivo de dispositivos IoT.

A implementação de fatiamento de rede (“network slicing”), característica essencial das redes 5G, adiciona complexidade aos mecanismos de RRM. O fatiamento permite que diferentes serviços e aplicações operem em fatias virtualizadas da mesma infraestrutura

física, cada uma com seus próprios requisitos de desempenho e isolamento. Esse modelo, embora poderoso, exige soluções sofisticadas para garantir a utilização justa dos recursos e o isolamento adequado do desempenho entre diferentes entidades envolvidas. Assim, torna-se necessário o desenvolvimento de abordagens avançadas para otimizar o uso dos recursos sob essas condições desafiadoras.

Entre os diversos fatores que influenciam o desempenho das redes 5G/6G, o consumo de energia dos dispositivos móveis é um aspecto crucial. Muitos dispositivos dependem de baterias com capacidade limitada, tornando a eficiência energética um dos principais objetivos em sua operação. Nesse cenário, o desenvolvimento de modelos capazes de estimar e otimizar o consumo de energia torna-se essencial para prolongar a vida útil das baterias e reduzir os custos operacionais. O uso de técnicas de aprendizado de máquina surge como uma alternativa promissora para resolver esse problema [1], permitindo a modelagem precisa do consumo energético com base em uma ampla gama de parâmetros.

Este trabalho tem como objetivo principal a modelagem e implementação de uma solução baseada em aprendizado de máquina para estimar o consumo de energia em dispositivos conectados às redes celulares. O modelo proposto utilizará parâmetros como a quantidade de recursos utilizados, potência de transmissão, potência de recepção e largura de banda. Ao estimar com precisão o consumo de energia, o modelo permitirá a otimização do uso da interface de rede, promovendo maior eficiência energética e contribuindo para o desenvolvimento de ambientes conectados mais sustentáveis.

O processo de análise dos dados envolveu a execução de várias etapas de pré-processamento. Inicialmente, realizamos a extração das séries temporais a partir de arquivos DLOG, que continham uma estrutura em árvore XML, responsável por descrever as características dos dados, juntamente com uma seção em formato binário, onde estavam armazenadas as medições temporais. Em seguida, aplicamos um filtro passa-baixa às séries temporais com o intuito de suavizar as variações, minimizando os ruídos de alta frequência e permitindo uma visualização mais precisa dos padrões subjacentes. Posteriormente, utilizamos a transformação wavelet, uma técnica avançada de análise de sinais, para detectar e localizar anomalias tanto no domínio da frequência quanto no domínio do tempo. Após a identificação dessas irregularidades, as anomalias foram removidas de forma eficiente, cortando-as diretamente e criando diversas novas séries temporais a partir dos fragmentos resultantes. Esse procedimento assegura a manutenção da qualidade dos dados e permite uma análise mais precisa e confiável para a aplicação dos modelos de aprendizado de máquina.

Dando continuidade ao processamento inicial, os dados foram refinados para permitir a aplicação de dois modelos distintos de aprendizado de máquina voltados para a previsão do consumo energético. O primeiro modelo, focado no pico de consumo, utilizou os padrões de curva extraídos das séries temporais segmentadas em intervalos de 10 milissegundos. A cada segmento, foi identificado o pico de energia, calculado como a média dos valores em torno do ponto de maior consumo. Essa abordagem simplificou o problema ao concentrar-se nos momentos de maior demanda energética.

Já o segundo modelo foi projetado para capturar toda a dinâmica da curva de consumo. Os dados das séries temporais foram novamente segmentados, e os valores de tempo  $x$  de cada segmento foram padronizados para uma escala uniforme, permitindo que o modelo aprendesse diretamente a relação entre  $x$  e  $y$ , os valores de consumo energético. Essa meto-

dologia possibilitou a previsão detalhada da curva de consumo, com maior granularidade e precisão.

Ambos os modelos foram treinados utilizando as séries temporais pré-processadas, garantindo que ruídos e anomalias previamente detectados não afetassem a acurácia das previsões. A aplicação dessas abordagens complementares não apenas resultou em previsões altamente precisas, mas também permitiu explorar diferentes aspectos do consumo energético, como o comportamento global da curva e os picos de demanda. Essa estratégia contribui diretamente para a otimização do uso de energia em dispositivos de redes celulares, alinhando-se aos desafios e objetivos das redes 5G/6G.

## 2 Metodologia

A metodologia está dividida em duas etapas principais: processamento de dados e treinamento de modelos. Essas etapas são fundamentais para transformar os dados brutos em um formato utilizável e desenvolver modelos capazes de prever o comportamento das séries temporais analisadas.

**Processamento de dados:** a etapa de processamento de dados tem como ponto de partida a extração de séries temporais a partir de arquivos no formato dlog. Esses arquivos contêm uma estrutura híbrida, composta por uma árvore XML, que organiza metadados e configurações, e dados binários, onde as medições das séries temporais são armazenadas. Para cada arquivo, foi extraída uma única série temporal, e os dados resultantes foram organizados em vários arquivos no formato CSV, com cada série armazenada separadamente.

Após a criação dos arquivos CSV, foi aplicado um filtro passa-baixa em cada série temporal para reduzir os ruídos presentes nos sinais. Essa etapa é essencial para garantir que as séries temporais contenham apenas as informações relevantes, minimizando distorções que poderiam afetar a análise e a modelagem subsequentes.

Com os dados filtrados, desenvolvemos um algoritmo de detecção de anomalias, projetado para identificar irregularidades nos sinais que poderiam comprometer a integridade dos modelos. Após a identificação das anomalias, essas regiões foram recortadas dos sinais originais, garantindo que os dados finais utilizados fossem limpos e representassem padrões consistentes.

**Treinamento de modelos:** na etapa de treinamento de modelos, foi projetada e implementada uma rede neural com a finalidade de prever a forma da curva resultante das operações de transmissão ou recepção. O modelo recebe como entrada as características das transmissões ou recepções, incluindo parâmetros de configuração do equipamento e condições operacionais. Como saída, a rede neural retorna a forma esperada da curva associada a essas operações.

Esse modelo foi desenvolvido para capturar as relações entre os parâmetros de entrada e o comportamento dinâmico do sinal, oferecendo uma ferramenta robusta para prever e analisar padrões de séries temporais em condições similares. Essa abordagem garante a eficácia no tratamento dos dados e na modelagem dos fenômenos analisados, permitindo que os resultados do trabalho sejam aplicados em contextos práticos e experimentais.

Com o consumo de corrente em mãos, é possível calcular a potência consumida mul-

tiplicando esse valor pela tensão utilizada nos experimentos (Tensão entregue pela bateria do dispositivo). Para este caso, o valor de voltagem é de 5v. Em seguida, o consumo de energia pode ser determinado através da área embaixo da curva de consumo de potência. Equivalentemente, a multiplicação do intervalo de tempo e a potência média nesse intervalo.

Assim, nossa abordagem foi focar no consumo da corrente, pois os demais fatores são fixos para o cálculo do consumo de energia.

## 2.1 Extração de dados

Os arquivos no formato dlog utilizados possuem uma estrutura dividida em duas partes principais: uma árvore XML e dados binários. A árvore XML contém informações de configuração e metadados, enquanto os dados binários representam as medições do consumo de corrente (série temporal).

A árvore XML organiza informações essenciais sobre o dispositivo e a configuração do experimento para a geração das series temporais:

- **Configurações do Canal:** Incluem níveis e inclinações de disparo para corrente e tensão, modos de medição (habilitação para tensão e corrente), e faixas de operação, com opções de ajuste automático. Além disso, o modelo do dispositivo e suas opções configuradas são descritos.
- **Parâmetros Temporais:** Contêm o tempo total de operação, deslocamentos temporais, e o tempo de integração (*tint*), que é usado para calcular o intervalo de amostragem.
- **Configurações da Interface Gráfica:** Detalham ajustes visuais, como ganho, deslocamento e habilitação de traços de tensão e corrente, além das posições de marcadores utilizados na análise visual.

Os dados binários armazenam as medições das séries temporais registradas pelo dispositivo. O intervalo de amostragem é calculado com base no parâmetro *tint* extraído do XML. A partir dessa informação, os timestamps foram gerados multiplicando o tempo de integração pelo índice de cada amostra, garantindo uma escala temporal precisa para as séries.

Após a extração e processamento, cada série temporal foi convertida em um arquivo CSV com duas colunas: a primeira contendo os timestamps e a segunda os valores medidos. Cada arquivo dlog gerou um CSV correspondente, mantendo o mesmo nome base, facilitando a rastreabilidade e o uso em análises posteriores.

Foi aplicado um filtro passa-baixa para remover os ruídos presentes nos sinais. O objetivo dessa etapa foi garantir que as séries temporais refletissem apenas as informações relevantes, eliminando componentes de alta frequência que poderiam interferir na análise e modelagem subsequentes.

O filtro passa-baixa foi implementado utilizando um filtro *Butterworth*, conhecido por sua resposta de magnitude suave e eficiência em eliminar ruídos indesejados. O processo foi realizado com as seguintes configurações:

- **Frequência de amostragem ( $f_s$ ):** 48.828 Hz, derivada da análise dos arquivos `dlog`.
- **Frequência de corte ( $f_c$ ):** 600 Hz, escolhida para manter as informações relevantes na faixa de baixa frequência através de tentativa e erro.
- **Ordem do filtro:** 2, balanceando uma atenuação suficiente dos ruídos sem comprometer a integridade do sinal.

A implementação foi baseada na biblioteca `scipy`, com o uso da função `butter` para definir os coeficientes do filtro e `filtfilt` para aplicar o filtro de forma bidirecional. O resultado do processo de filtragem foi um conjunto de arquivos `CSV` contendo séries temporais suavizadas, prontas para análise e treinamento de modelos. A preservação das características principais do sinal e a remoção de ruídos garantiram uma base sólida para as etapas seguintes, como a detecção de anomalias e a modelagem. O resultado desse processo pode ser visto nas figuras 1 e 2. Na figura 1, podemos visualizar o comportamento da séries temporal antes da aplicação do filtro passa-baixa, enquanto na figura 2, podemos visualizar a série temporal depois da aplicação do filtro passa-baixa.

Essa etapa foi essencial para assegurar que os modelos de aprendizado de máquina recebessem dados de alta qualidade, contribuindo para a precisão e robustez dos resultados.

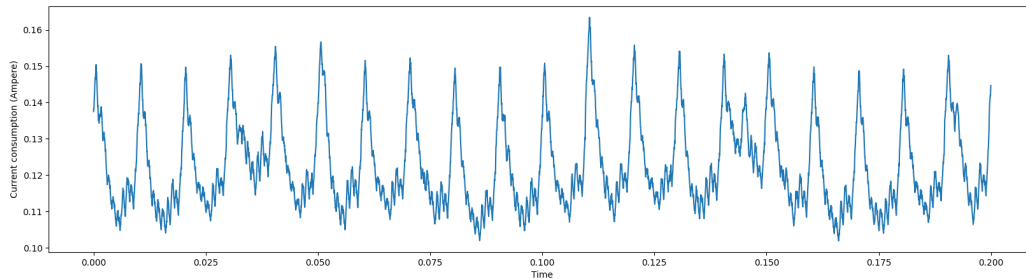


Figura 1: Sinal antes do filtro passa-baixa.

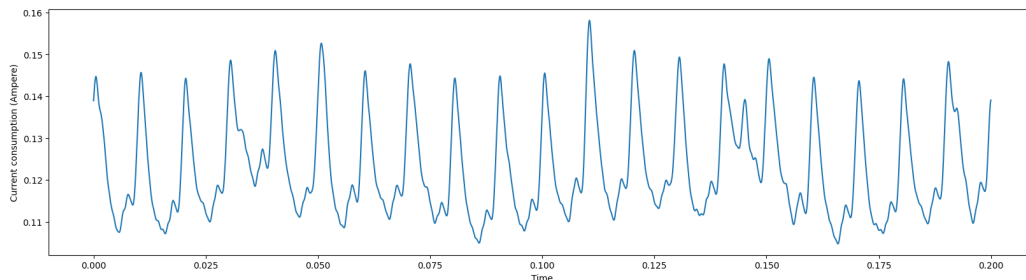


Figura 2: Sinal depois do filtro passa-baixa.

## 2.2 Remoção de anomalia

Mesmo depois da suavização, ainda tinha casos de erro de medida durante as séries temporais. Para detectar esses erros de medidas (anomalias), utilizamos uma transformada de wavelet [9]. A transformada de wavelet é uma ferramenta matemática amplamente utilizada para a análise de sinais [8], permitindo a representação de um sinal no domínio tempo-frequência. Diferentemente da Transformada de Fourier, que representa o sinal exclusivamente em termos de frequências globais, a Transformada de Wavelet fornece uma visão local das frequências do sinal, possibilitando a detecção de variações em regiões específicas. Para realizar essa transformação, a transformada de wavelet decompõe um sinal em uma série de funções base chamadas **wavelets**, que são geradas a partir de uma função mãe ( $\psi(t)$ ) por operações de dilatação e translação. Matematicamente, a Transformada de Wavelet Contínua (CWT) de um sinal  $x(t)$  é dada por:

$$W(a, b) = \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-b}{a}\right) dt,$$

onde:

- $a$  é o parâmetro de escala, que controla a dilatação ou compressão da wavelet, determinando a resolução em frequência.
- $b$  é o parâmetro de translação, que desloca a wavelet ao longo do tempo, permitindo a localização temporal.
- $\psi^*(t)$  é a função wavelet mãe conjugada complexa.

A operação de dilatação e translação permite que a Transformada de Wavelet analise o sinal em diferentes escalas, correspondendo a diferentes faixas de frequência, enquanto preserva a localização no tempo. Isso resulta em um mapa tempo-frequência que pode ser usado para identificar padrões ou anomalias no sinal.

As anomalias em um sinal geralmente correspondem a variações abruptas ou não esperadas que podem ser localizadas no domínio do tempo e caracterizadas por componentes de alta frequência. A Transformada de Wavelet é especialmente útil nesse contexto devido às seguintes propriedades:

- **Localização temporal:** A wavelet fornece informações sobre o momento exato em que ocorre uma variação significativa no sinal, como um pico ou uma descontinuidade. Isso é essencial para identificar anomalias localizadas.
- **Análise multiescala:** A capacidade de analisar o sinal em diferentes escalas permite isolar componentes de alta frequência associadas a ruídos ou transições abruptas, separando-os das componentes de baixa frequência que representam o comportamento geral do sinal.
- **Detecção automática de padrões:** A análise do coeficiente de wavelet pode revelar mudanças abruptas nos valores, como picos ou desvios súbitos, que são indicadores típicos de anomalias.

Com isso em mente, cada período do sinal pode ser caracterizado pelos coeficientes de wavelet associados às diferentes escalas e posições temporais, que indicam a contribuição das frequências presentes naquele intervalo específico. Esses coeficientes fornecem informações valiosas sobre as variações de frequência ao longo do tempo, possibilitando a identificação de padrões e irregularidades.

Para analisar um período específico, calculam-se os coeficientes de wavelet ao longo das escalas para o intervalo de interesse. Esses coeficientes representam a energia do sinal em diferentes bandas de frequência naquele período, oferecendo uma forma compacta de caracterizar a dinâmica do sinal em termos de frequência. Ao acumular esses coeficientes para todos os períodos da série temporal, é possível calcular métricas estatísticas como a média e o desvio padrão dos valores dos coeficientes.

A média dos coeficientes é obtida pela soma de todos os valores dos coeficientes em todos os períodos, dividida pelo número total de valores. Já o desvio padrão é calculado considerando o grau de dispersão dos coeficientes em relação à média, sendo uma medida da variabilidade dos valores ao longo do sinal. Essas métricas estatísticas são úteis para estabelecer uma base de comparação que ajuda a identificar desvios significativos.

Com a média e o desvio padrão em mãos, é possível categorizar períodos como anômalos se os coeficientes associados a esses períodos ultrapassarem certos limites estabelecidos em relação ao desvio padrão. Neste caso, períodos cujos coeficientes apresentem valores acima ou abaixo de um intervalo definido, como  $[Média - 2 \times \text{Desvio Padrão}, Média + 2 \times \text{Desvio Padrão}]$ , foram classificados como anômalos. Esses desvios podem indicar eventos excepcionais, como picos abruptos, mudanças de tendência ou descontinuidades no sinal como pode ser visualizado na figura 3.

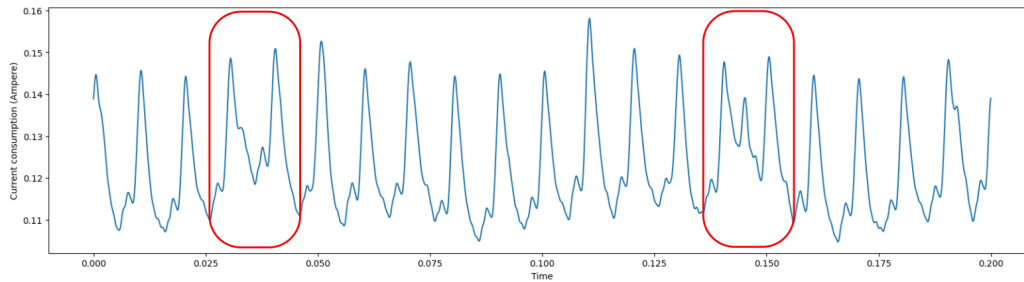


Figura 3: Anomalias no sinal.

## 2.3 Modelos

Neste projeto, utilizou-se um dataset com labels específicas: largura de banda, potência de transmissão, potência de recepção e dados representando uma curva de consumo. A partir dessas informações, foram desenvolvidos dois modelos de aprendizado de máquina, usando PyTorch [4], para previsão de consumo de energia, cada um com abordagens distintas para interpretar os dados e os padrões de consumo, porém com o mesmo objetivo.

O Modelo 1 baseia-se na identificação do pico de corrente consumido nos dados disponíveis. Este pico é utilizado como uma referência para criar uma relação direta entre



os dados do dataset e as labels, simplificando a previsão ao focar no momento de maior consumo. Por outro lado, o Modelo 2 adota uma abordagem mais abrangente. Ele aprende a curva de consumo utilizando todos os dados do dataset em conjunto com as labels. A curva aprendida é então utilizada para calcular o consumo como no modelo 1, utilizando o pico da curva.

### 2.3.1 Modelo 1 - processamento de dados

Para preparar o dataset para o Modelo 1, utilizamos um conjunto de medições de consumo de corrente com diferentes valores de potência de transmissão e recepção. Essas medições formam uma curva contínua, na qual o padrão se repete a cada 10 milissegundos (um quadro em tecnologias de comunicação celular), com valores ligeiramente diferentes, mas mantendo as mesmas características gerais.

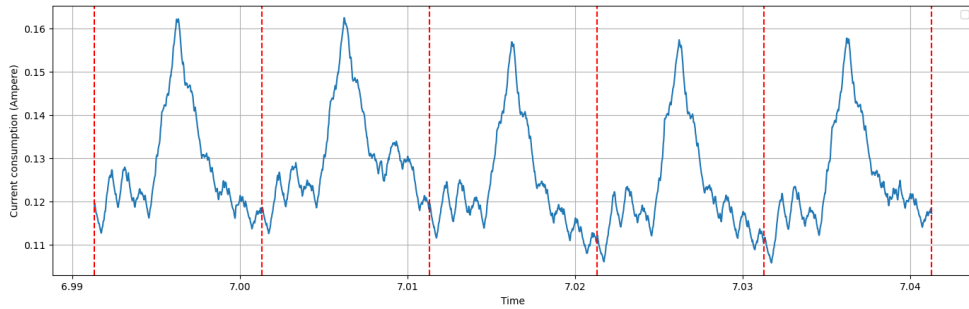


Figura 4: Corte do padrão de curva

O pré-processamento consistiu em dividir a curva contínua em segmentos de 10 milissegundos, correspondendo a cada padrão de curva com os respectivos labels. Para cada segmento, foi identificado o pico da curva, definido como o maior valor de  $y$  (consumo de energia) no intervalo. A fim de garantir uma estimativa estável para o consumo de energia, foi calculada a média dos valores de  $y$  em uma janela de aproximadamente 0,5 milissegundos ao redor do pico. O valor resultante foi atribuído como o consumo de energia para aquele segmento, criando uma nova coluna com as informações processadas.

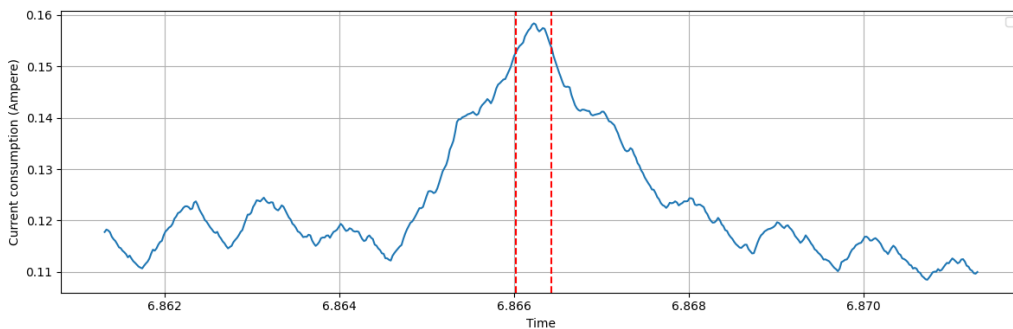


Figura 5: Intervalo dos valores do cálculo do pico de energia

### 2.3.2 Modelo 1 - estrutura modelo

O Modelo 1 é uma rede neural projetada para prever o consumo de energia com base em labels de potência de transmissão, potência de recepção e o consumo calculado no pré-processamento. Sua estrutura e funcionamento podem ser descritos da seguinte forma:

- **Camadas de Entrada:** A entrada do modelo é um vetor unidimensional, contendo os labels de entrada codificados como vetores one-hot [5]. Isso permite ao modelo processar informações categóricas de forma eficiente.
- **Camadas Ocultas:** A arquitetura inclui três camadas densas (fully connected) [6], intercaladas com a função de ativação ReLU para introduzir não-linearidade e um dropout com uma taxa de 20% é aplicado após a primeira camada para prevenir o overfitting e melhorar a generalização do modelo.:
  - **Entrada** → **64** → **128** → **64** → **Saída**
- **Camada de Saída:** A última camada reduz a dimensionalidade para um único neurônio, que representa o valor previsto de consumo de energia em formato contínuo.
- **Função de Perda:** Utiliza a Mean Squared Error (MSE) como métrica de erro, ideal para o problema de regressão contínua.
- **Otimizador:** O otimizador Adam, com taxa de aprendizado de 0,001, é empregado para ajustar os pesos do modelo durante o treinamento.
- **Treinamento:** O modelo é treinado por cinco épocas, utilizando um DataLoader com mini-batches de tamanho 5.

### 2.3.3 Modelo 2 - processamento de dados

O processamento dos dados para o Modelo 2 foi desenvolvido para permitir que o modelo aprendesse diretamente a curva de consumo energético. Inicialmente, a curva contínua foi dividida em segmentos de 10 milissegundos, correspondendo aos padrões repetitivos identificados no dataset como no modelo 1. Cada segmento foi tratado como uma nova instância com os mesmos labels associados.

Para garantir a padronização e facilitar o aprendizado do modelo, os valores de  $x$ (tempo) dentro de cada segmento foram substituídos por uma sequência padronizada de índices inteiros, começando em 0 e indo até 485, correspondente ao número de pontos em cada curva. Essa padronização eliminou a influência dos valores originais de  $x$ , permitindo que o modelo focasse no comportamento relativo da curva dentro de cada padrão.

Os dados pré-processados foram então organizados no formato de pares  $[(x_1, y_1), (x_2, y_2), \dots]$ , com os valores de  $x$  padronizados e os valores de  $y$ (consumo energia) preservando os valores reais da curva. Este formato preparou o dataset para que o Modelo 2 pudesse aprender a prever os valores de  $y$  com base na posição relativa  $x$  em cada segmento, capturando o comportamento estrutural das curvas de consumo ao longo dos padrões do dataset.

### 2.3.4 Modelo 2 - estrutura

O Modelo 2 é uma rede neural projetada para aprender diretamente a curva de consumo energético. Sua estrutura é composta pelos seguintes elementos:

- **Camadas de Entrada:** Cada entrada consiste em um vetor unidimensional que representa os valores de  $x$  padronizados para cada segmento da curva, combinados com os labels associados. Esses labels, como potência de transmissão e recepção, são codificados no formato one-hot.
- **Camadas Ocultas:** A rede neural possui uma sequência de sete camadas densas (fully connected), projetadas para extrair padrões relevantes dos dados e a função de ativação ReLU é usada em cada camada oculta, garantindo a capacidade de modelar relações não lineares nos dados:
  - **Entrada**  $\rightarrow$  **64**  $\rightarrow$  **128**  $\rightarrow$  **256**  $\rightarrow$  **512**  $\rightarrow$  **256**  $\rightarrow$  **128**  $\rightarrow$  **Saída**
- **Camada de Saída:** A camada final é composta por um único neurônio, que produz a previsão do valor de  $y$ ), correspondente ao consumo de energia no ponto específico da curva.
- **Função de Perda:** Utiliza a Mean Squared Error (MSE) como métrica de erro, apropriada para a tarefa de prever valores contínuos em um problema de regressão.
- **Otimizador:** Adota o otimizador Adam, configurado com uma taxa de aprendizado de 0,001, para ajustar os pesos da rede durante o treinamento.
- **Treinamento:** O modelo é treinado por cinco épocas, utilizando um DataLoader com batches de tamanho 100.

Esse modelo prediz a curva de consumo de diferentes valores de transmissão, recepção e largura de banda e para calcular o pico de consumo é utilizado o mesmo método do processamento de dados do modelo 1.

## 3 Resultados

Para avaliar o desempenho dos modelos desenvolvidos, foram utilizadas as seguintes métricas de regressão [7], que permitem mensurar a precisão e a qualidade das previsões:

- **Mean Absolute Error (MAE):** O MAE calcula a média dos valores absolutos dos erros entre as previsões  $\hat{y}_i$  e os valores reais  $y_i$ . Ele fornece a magnitude média dos erros, sem considerar sua direção. A fórmula é:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

Um MAE menor indica previsões mais próximas dos valores reais.

- **Mean Squared Error (MSE):** O MSE avalia o erro médio ao quadrado, penalizando erros maiores de forma mais severa. Sua fórmula é:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

Valores mais baixos de MSE indicam uma menor discrepância entre os valores previstos e reais.

- **Root Mean Squared Error (RMSE):** O RMSE é a raiz quadrada do MSE, representando o erro médio na mesma escala dos dados originais. Ele é definido como:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$

Assim como o MSE, o RMSE penaliza mais severamente os grandes desvios, mas fornece uma interpretação mais intuitiva em termos da unidade dos dados.

- **R-squared ( $R^2$ ):** O R-squared mede a proporção da variância nos valores reais que é explicada pelo modelo, indicando o quão bem o modelo ajusta os dados. Ele é calculado como:

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Onde  $\bar{y}$  é a média dos valores reais. Um R-squared próximo de 1 indica um excelente ajuste do modelo.

### 3.1 Resultados - modelo 1

Metric	Value
Mean Absolute Error (MAE)	0.0135
Mean Squared Error (MSE)	0.0010
Root Mean Squared Error (RMSE)	0.0317
R-squared ( $R^2$ )	0.9428

Tabela 1: Resultado das métricas no modelo 1

Os resultados para o consumo total mostram que o modelo obteve um bom desempenho, com R-squared de 0,9428 e erros moderadamente baixos (MAE: 0,0135, RMSE: 0,0317). Isso demonstra que o modelo conseguiu prever o pico de consumo de energia de cada padrão de curva com boa precisão, capturando as variações nos valores máximos ao longo do dataset.

A seguir, apresentamos três plots que mostram os picos de energia de três diferentes padrões de curva extraídos do dataset. Em cada gráfico, o valor previsto pelo modelo é destacado, evidenciando a proximidade das previsões com os valores reais do consumo máximo.

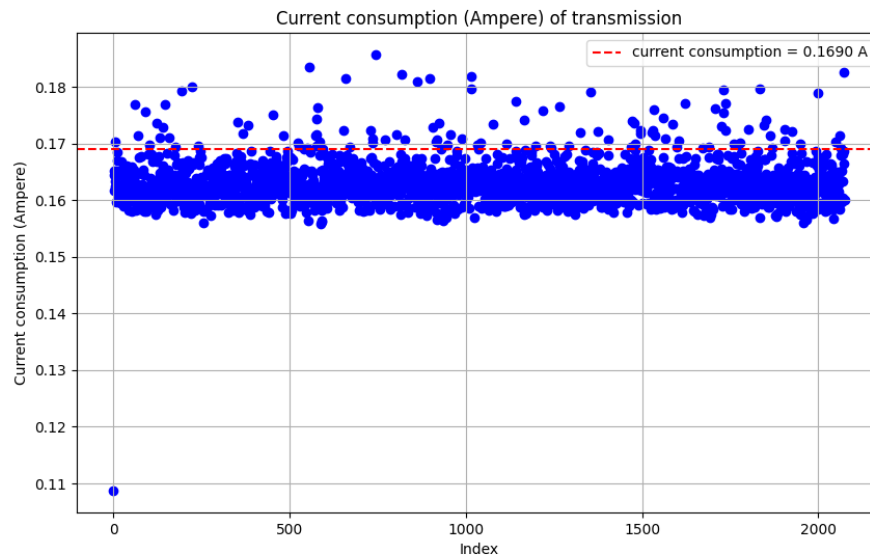


Figura 6: Padrão de transmissão com o valor predito: 0.1690 A

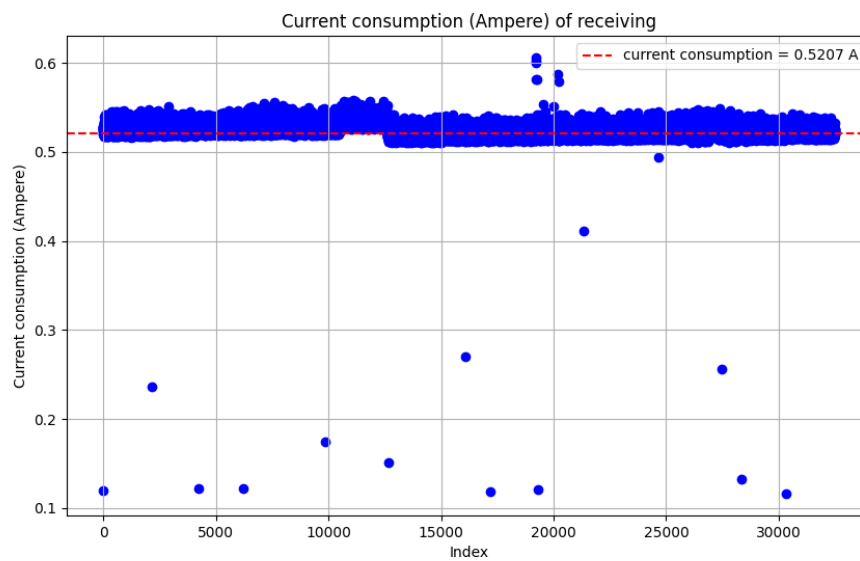


Figura 7: Padrão de recepção com o valor predito: 0.5207 A

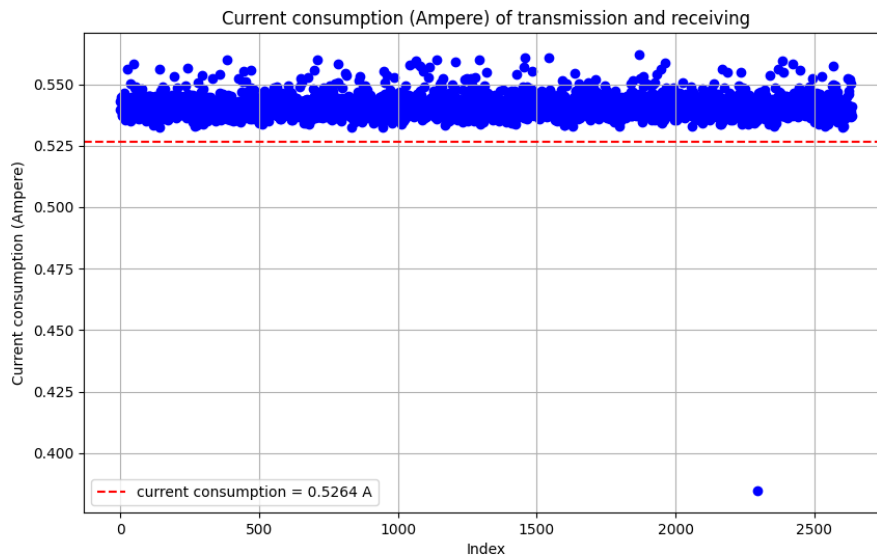


Figura 8: Padrão de transmissão e recepção com o valor predito: 0.5264 A

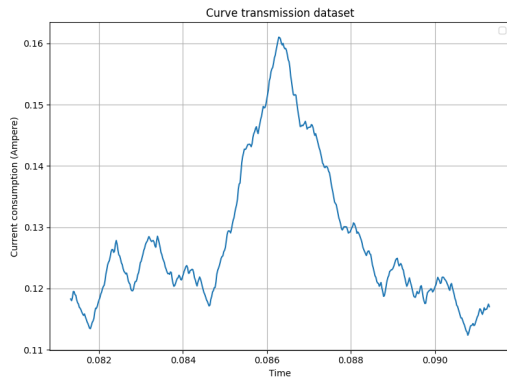
### 3.2 Resultados - modelo 2

Metric	Value
Mean Absolute Error (MAE)	0.0065
Mean Squared Error (MSE)	0.0002
Root Mean Squared Error (RMSE)	0.0147
R-squared ( $R^2$ )	0.9450

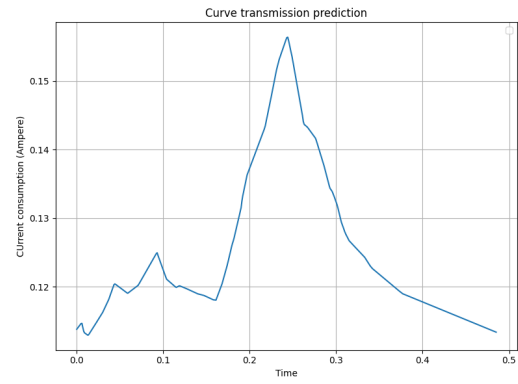
Tabela 2: Resultado das métricas no modelo 2

Os resultados obtidos para a previsão da curva de consumo indicam que o modelo teve um desempenho excelente, com um R-squared de 0,9450 e erros médios baixos (MAE: 0,0065, RMSE: 0,0147). Isso demonstra que o modelo conseguiu capturar com precisão o padrão repetitivo das curvas no dataset, reproduzindo fielmente o comportamento dos valores reais.

A seguir, são apresentados três plots que mostram as previsões realizadas pelo modelo em três padrões distintos da curva de consumo, e três plots da curva equivalente do dataset. É possível notar que o modelo é capaz de reproduzir com grande similaridade os padrões contínuos do dataset.

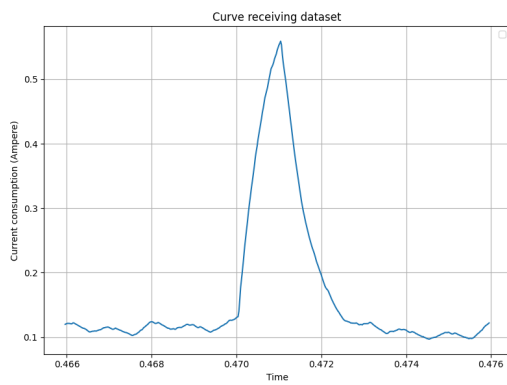


(a) Dataset

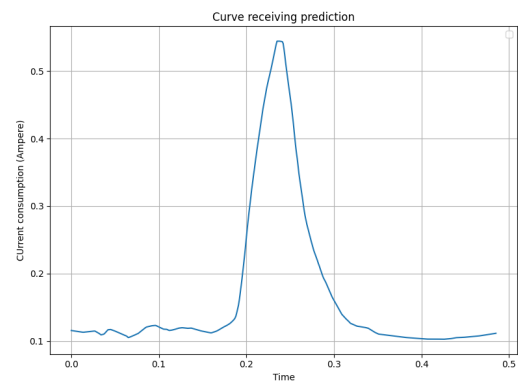


(b) Predição

Figura 9: Curva de transmissão do dataset e predito

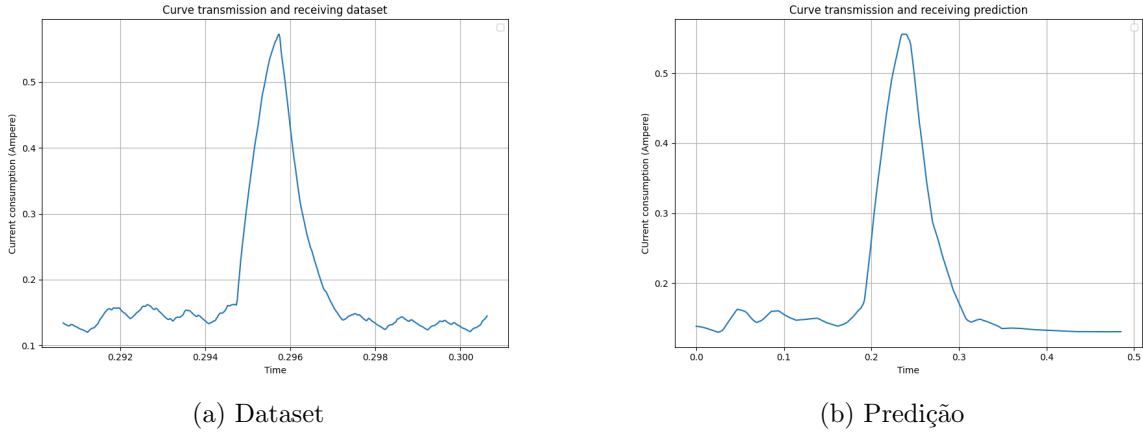


(a) Dataset



(b) Predição

Figura 10: Curva de recepção do dataset e predito



(a) Dataset (b) Predição

Figura 11: Curva de transmissão e recepção do dataset e predito

Com esse modelo calculamos o consumo da curva prevista usando o método do processamento de dado do modelo 1 encontrando o seguinte resultado:

Metric	Value
Mean Absolute Error (MAE)	0.0138
Mean Squared Error (MSE)	0.0009
Root Mean Squared Error (RMSE)	0.0296
R-squared ( $R^2$ )	0.9500

Tabela 3: Resultado das métricas no calculo de consumo usando o modelo 2

Os resultados obtidos para a previsão da curva de consumo com ajustes também foram excelentes, com um R-squared de 0,9500, MAE de 0,0138 e RMSE de 0,0296. Esse desempenho destaca a capacidade do modelo em prever não apenas a forma da curva, mas também o consumo máximo de energia, com alta precisão.

Nos três plots apresentados a seguir, mostramos o consumo real e previsto para três padrões distintos do dataset. Além disso, a curva prevista foi utilizada para calcular o pico de energia, da mesma forma que foi calculado no modelo 1 e este valor foi comparado com o consumo de energia real do dataset.



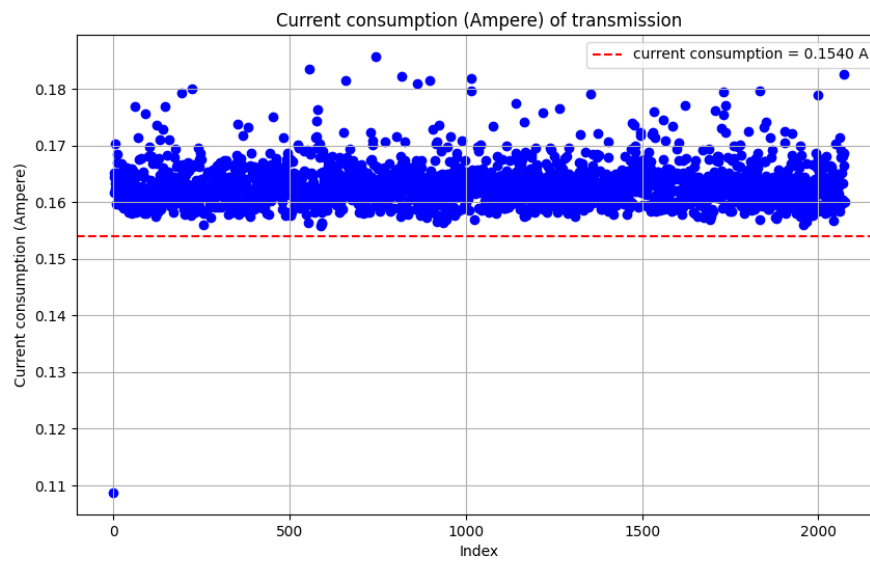


Figura 12: Padrão de transmissão com o valor predito do modelo 2: 0.1540

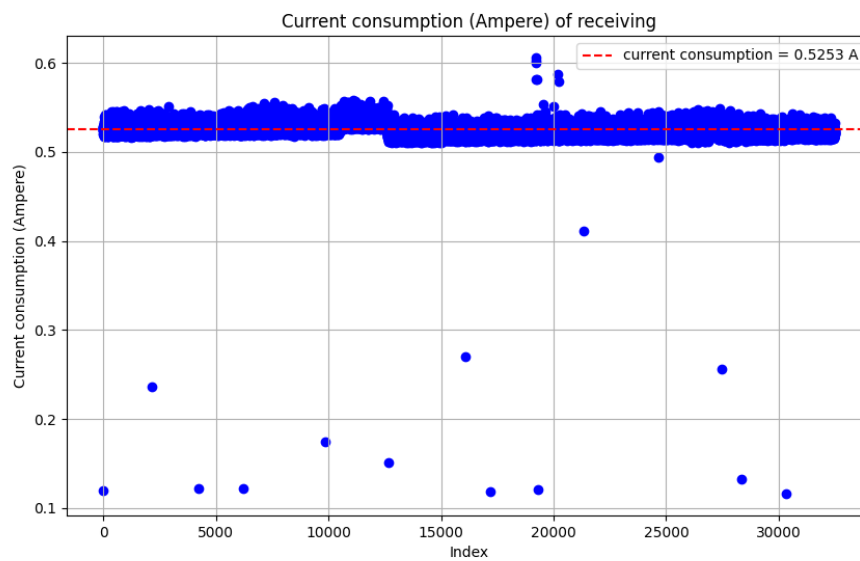


Figura 13: Padrão de recepção com o valor predito do modelo 2: 0.5253

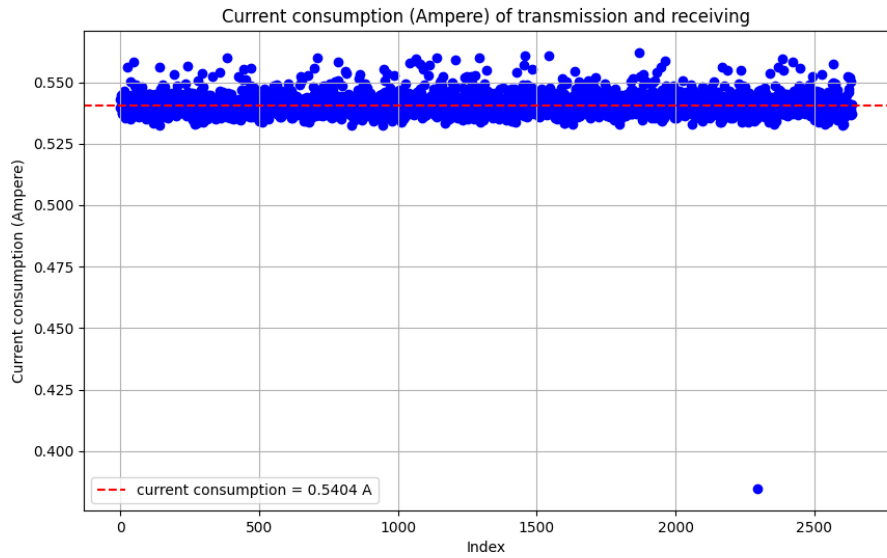


Figura 14: Padrão de transmissão e recepção com o valor predito do modelo 2: 0.5404

## 4 Conclusões

Neste projeto, desenvolvemos e avaliamos dois modelos distintos de aprendizado de máquina para a previsão de consumo energético utilizando um dataset com padrões contínuos de curva de consumo. O Modelo 1 baseou-se na identificação do pico de consumo em cada segmento da curva, simplificando a tarefa preditiva para focar nos momentos de maior consumo. Por outro lado, o Modelo 2 adotou uma abordagem mais abrangente, aprendendo diretamente os padrões completos da curva de consumo e, em seguida, extraíndo informações relevantes, como o pico de potência.

Os resultados demonstraram que ambos os modelos obtiveram desempenhos satisfatórios nas tarefas propostas, com métricas de precisão elevadas. O Modelo 2, particularmente no cenário ajustado da curva de consumo, apresentou o melhor desempenho geral, alcançando um R-squared de 0,9500, o que reflete sua capacidade de capturar de forma precisa as variações estruturais do dataset e prever tanto a forma da curva quanto o consumo máximo de energia com alta fidelidade.

Esses resultados destacam a eficácia do aprendizado de padrões contínuos para problemas preditivos relacionados a consumo energético, especialmente em cenários onde o padrão de curva contém informações ricas e recorrentes. Enquanto o Modelo 1 oferece uma solução mais simples e eficiente para casos em que o foco está nos picos de consumo, o Modelo 2 é mais adequado para aplicações que exigem maior granularidade e compreensão da forma completa da curva.

A integração desses dois modelos pode ser uma direção futura interessante, combinando a simplicidade do cálculo de picos com a robustez da previsão completa da curva. Além disso,

investigações adicionais podem incluir a incorporação de dados externos, como variáveis ambientais ou de carga, para aprimorar ainda mais a precisão dos modelos. Este trabalho demonstra a importância de escolher a abordagem certa para cada necessidade e fornece uma base sólida para estudos futuros em previsão de consumo energético.

## Referências

- [1] M. Lauridsen, L. Noel, T. B. S rensen, P. Mogensen, *An Empirical LTE Smartphone Power Model with a View to Energy Efficiency Evolution*, Intel®Technology Journal 18 (1) (2014) 172193.
- [2] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley (1901).
- [3] D. E. Knuth and L. Lamport, *A structural analysis of the role of gnus and gnats in the post-modernistic, crypto-existential Weltanschauung of neo-liberal Tibeto-Vietnamese leaf blower operators as manifest in the sexual symbology of the Los Angeles Phone Directory*. Journal of Gnu Technology, **23** (6), 12–87 (March 1996).
- [4] Pytorch, *PyTorch documentation*, <https://pytorch.org/docs/stable/index.html>, (Accessed: Dec 13, 2024).
- [5] Alice Zheng, Amanda Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O’Reilly Media (2018).
- [6] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Jonathan Taylor, *An Introduction to Statistical Learning: with Applications in Python*. Springer (2023).
- [7] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor-Flow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media (2019).
- [8] Kun Wu, Lei Zhu, Weihang Shi, Wenwu Wang, Jin Wu *Self-Attention Memory-Augmented Wavelet-CNN for Anomaly Detection*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 3, pp. 1374-1385, March 2023.
- [9] O. Rioul and M. Vetterli, *Wavelets and signal processing*. IEEE Signal Processing Magazine, vol. 8, no. 4, pp. 14-38, Oct. 1991.