

# Predição do Consumo Energético de Dispositivos LoRa usando Aprendizagem de Máquina

Henrique P. P. dos S. Pimentel\* Hilton de A. Viana\*\*  
Adriano S. de Albuquerque\*\* Danilo R. B. de Araújo\*

\* Departamento de Computação (DC) Universidade Federal Rural de  
Pernambuco (UFRPE) – Recife-PE  
(e-mail:henrique.pablo,danilo.araujo@ufrpe.br)

\*\* Programa de Pós-Graduação em informática Aplicada (PPGIA)  
Universidade Federal Rural de Pernambuco (UFRPE) – Recife-PE,  
(e-mail: hilton.aquinoviana,adriano.severiano@ufrpe.br)

---

**Abstract:** The Internet of Things (IoT) is a constantly evolving concept that has gained prominence in both the academic community and industry. Within it, energy consumption is a fundamental factor in determining the operating time of devices and the frequency necessary to carry out their maintenance. This paper investigates the application of machine learning algorithms to predict the energy consumption of IoT-LoRa devices, allowing estimation of the devices' battery life and autonomy. The methodology considered the creation of a data set based on experiments with Event stream processing (ESP32) development boards, capturing metrics such as sleep time, connection type and energy consumption. Artificial Intelligence (AI) techniques are then applied to predict energy consumption based on these variables. According to the results obtained, the best technique for predicting energy consumption is Decision Tree, with a coefficient of determination greater than 96%. The study contributes to decision-making processes that aim to select IoT devices considering the autonomy designed for the batteries of such devices.

**Resumo:** A Internet das Coisas (IoT) é um conceito em constante evolução que tem conquistado destaque tanto na comunidade acadêmica quanto na indústria. Dentro dela, o consumo energético é um fator fundamental para determinar o tempo de funcionamento dos dispositivos e a frequência necessária para realizar a manutenção deles. Este artigo investiga a aplicação de algoritmos de aprendizado de máquina para predição do consumo energético de dispositivos IoT-LoRa, permitindo estimar a duração da bateria dos dispositivos e sua autonomia. A metodologia considerou a criação de um conjunto de dados a partir de experimentos com placas de desenvolvimento *Event stream processing* (ESP32), capturando métricas como tempo de hibernação, tipo de conexão e consumo energético. Técnicas de Inteligência Artificial (IA) são então aplicadas para prever o consumo energético com base nessas variáveis. De acordo com os resultados obtidos, a melhor técnica para prever o consumo energético é a *Decision Tree*, com um coeficiente de determinação superior a 96%. O estudo contribui para processos decisórios que visam selecionar dispositivos IoT considerando a autonomia projetada para as baterias de tais dispositivos.

**Keywords:** Internet of Things (IoT); Energy Consumption; Artificial Intelligence (AI); Optimization.

**Palavras-chaves:** Internet das Coisas; Consumo energético; Inteligência Artificial; Otimização.

---

## 1. INTRODUÇÃO

A evolução tecnológica tem promovido transformações significativas em diversos setores, com a Internet das Coisas (IoT) emergindo como uma força disruptiva. Este conceito, conforme descrito por Laghari et al. (2021), refere-se a uma rede complexa de objetos inteligentes que se comunicam e interagem tanto entre si quanto com o ambiente. A integração de dispositivos IoT abre novas possibilidades para otimizar processos, coletar dados e aprimorar a eficiência em uma ampla gama de aplicações,

desde o cotidiano doméstico até áreas mais complexas como a indústria e a agricultura.

No cerne da IoT está a conexão entre dispositivos e objetos, viabilizada por uma variedade de tecnologias de comunicação, dentre as quais Ahsan et al. (2021) cita Wi-Fi, *Bluetooth*, *Zigbee* e *LoRaWAN*. Cada contexto de aplicação demanda considerações específicas ao escolher a tecnologia de conexão mais adequada, levando em conta requisitos como alcance, consumo de energia, taxa de transferência de dados e custo, conforme destacado por Callebaut et al. (2021).

Este artigo investiga qual algoritmo de aprendizado de máquina é mais adequado para estimar o consumo energético Reddy Maddikunta et al. (2020) de placas de comunicação e desenvolvimento ESP32 com foco na comunicação por meio da tecnologia LoRa (*Long Range*). O objetivo é facilitar o processo decisório dos envolvidos quando o tempo de duração das baterias for um elemento crítico para análise dos equipamentos mais adequados.

O artigo está estruturado da seguinte forma. Na Seção 2 apresenta-se a fundamentação teórica para fornecer uma visão geral sobre os algoritmos utilizados. Na Seção 3 encontra-se a metodologia, detalhando como a pesquisa foi conduzida. A Seção 4 aborda os resultados obtidos a partir dos algoritmos de *Machine Learning* (ML) aplicados sobre uma base de dados construída na pesquisa. Por fim, é fornecida uma seção com as conclusões do trabalho e sugestões para trabalhos futuros.

### 1.1 Trabalhos relacionados

Nos últimos anos, o uso de dispositivos IoT tem ganhado cada vez mais destaque, abrangendo diversas áreas de aplicação e oferecendo várias vantagens. Entre essas vantagens, destaca-se a longevidade da bateria e os meios para maximizá-la. Reddy Maddikunta et al. (2020) ilustram que uma das abordagens para preservar a vida útil da bateria é por meio da predição usando modelos de aprendizado de máquina.

Considerando a viabilidade da implementação de modelos de aprendizado de máquina, Bhattacharya et al. (2021) examinaram o uso de *Deep Learning* para prever a vida útil da bateria. Eles sugerem que modelos mais complexos, dependendo da quantidade e qualidade dos dados utilizados, podem apresentar um desempenho superior em comparação com modelos de aprendizado de máquina mais simples.

Um desafio comum no uso de dispositivos IoT é sua dependência de baterias, que tendem a deteriorar-se ao longo do tempo. Portanto, é crucial explorar estratégias para minimizar a degradação da bateria e, assim, prolongar a vida útil das redes IoT. Ergun et al. (2019) propuseram diminuir a degradação da bateria usando *Model Predictive Control* (MPC), demonstrando que isso pode melhorar a vida útil da bateria em até 68,5% em comparação com modelos lineares simples.

Dispositivos IoT tendem a ter um consumo de energia relativamente baixo, pois são alimentados por baterias. É importante estudar meios de reduzir os custos de manutenção, ou seja, a substituição dessas baterias. Um exemplo disso é o uso de energia solar, cuja previsão é essencial para uma gestão e utilização eficiente dos recursos disponíveis. Kraemer et al. (2017) abordaram o uso de diferentes métodos de aprendizado de máquina para prever a energia solar em um ambiente de sensores restritos, com base em dados meteorológicos publicamente disponíveis. Seus resultados indicaram que é possível prever a energia solar mesmo com acesso limitado aos dados, melhorando progressivamente à medida que o sistema opera.

Para determinar o consumo de energia e a vida útil da bateria, é possível utilizar simuladores, como demonstrado por Faye et al. (2022), que usaram a ferramenta LoRaSIM

para simular diferentes cenários com diferentes confirmações de recebimento após o envio de um pacote. Seus resultados mostraram que o consumo de energia da rede aumenta significativamente dependendo do estado dos dispositivos finais.

Embora possam ser citados diversos trabalhos com a preocupação de avaliar a vida útil de baterias por meio da aprendizagem de máquina, há uma carência de estudos focados em infraestruturas de comunicação e, em especial, em LoRaWAN. Embora o estudo Faye et al. (2022) tenha preocupação com a tecnologia LoRa, os dados usados são provenientes de simulações computacionais e não obtidos experimentalmente. Logo, este artigo avalia o uso de ML para prever o tempo de vida útil de dispositivos LoRa a partir de dados reais, coletados experimentalmente.

## 2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, será fornecido um breve resumo das técnicas de ML usadas no estudo.

### 2.1 Multi-Layer Perceptron

Conforme explicado por Liu et al. (2021), *Multi-Layer Perceptron* (MLP) é uma arquitetura de rede neural artificial *feedforward*. Este modelo é composto por, pelo menos, três camadas de nós: uma de entrada; uma ou mais camadas ocultas; e uma de saída. Cada nó é um *perceptron* simples que utiliza uma função de ativação não-linear para processar os dados.

Durante o processo de treinamento, os dados de entrada são propagados através da rede, passando pelas camadas ocultas, onde são transformados e ajustados. O treinamento é realizado utilizando algoritmos como *backpropagation*, que atualizam os pesos da rede para minimizar o erro entre as saídas previstas e os valores reais Naskath et al. (2023).

### 2.2 Decision Tree

As *Decision Trees*, também conhecidas como Árvores de decisão são modelos de aprendizado supervisionado que são utilizados para classificação e regressão. Elas dividem o espaço de entrada em regiões retangulares e associam uma classe ou valor de regressão a cada região.

A construção da árvore é realizada de forma recursiva, dividindo o conjunto de dados em subconjuntos menores com base em características específicas. O objetivo é minimizar a impureza dos subconjuntos, geralmente medida pela entropia ou índice Gini Charbuty and Abdulazeez (2021).

### 2.3 Random Forest

O *Random Forest* é uma extensão das Árvores de Decisão que utiliza múltiplas árvores de decisão durante o treinamento. Para problemas de classificação, o modelo escolhe a classe mais frequente entre todas as árvores. Para problemas de regressão, ele calcula a média das previsões das árvores individuais Schonlau and Zou (2020).

Durante o treinamento, várias árvores de decisão são construídas utilizando subconjuntos aleatórios do conjunto de dados e características. As previsões são então agregadas para obter uma saída final mais robusta e precisa, reduzindo o risco de *overfitting* e melhorando a capacidade de generalização do modelo.

#### 2.4 K-Nearest Neighbors

Conforme explicado por Kohli et al. (2020), *K-Nearest Neighbors* (KNN) é um algoritmo de aprendizado de máquina baseado em instâncias que classifica objetos com base nas classes dos exemplos mais próximos no espaço de características.

Para classificar um novo exemplo, o algoritmo calcula a distância entre o exemplo e todos os outros exemplos no conjunto de treinamento. Em seguida, seleciona os K exemplos mais próximos (vizinhos) e atribui a classe mais comum entre eles ao novo exemplo.

#### 2.5 Linear Regression

*Linear Regression* (LR), também conhecida como Regressão Linear, é um modelo estatístico que examina a relação linear entre duas ou mais variáveis. É usado para prever um valor numérico com base em variáveis independentes.

Conforme lembrado por Kohli et al. (2020), o modelo de regressão linear ajusta uma linha reta aos dados, minimizando a soma dos quadrados das diferenças entre os valores observados e os valores previstos pela linha. A equação da linha, onde  $y$  é a variável dependente,  $x_1, x_2, \dots, x_n$  são as variáveis independentes,  $\beta_0$  é o intercepto e  $\beta_1, \beta_2, \dots, \beta_n$  são os coeficientes que representam a mudança esperada, é geralmente expressa como:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (1)$$

### 3. METODOLOGIA

A metodologia da pesquisa considerou algumas macro-etapas. Na sequência, foi definido um arranjo experimental para coleta dos dados a serem usados. Os experimentos foram conduzidos utilizando placas, incluindo a medição de vários parâmetros, como por exemplo, o tempo de vida útil da bateria, e culminando na construção de um *dataset*. Na sequência, foram implementadas as técnicas de Inteligência Artificial (IA) consideradas na investigação, a saber: *Linear Regression* (LR), *Multilayer Perceptron*, *Decision Tree* (DT), *Random Forest* (RF) e *K-Nearest Neighbors*. Este artigo adotou uma abordagem quali-quantitativa para avaliar tanto o contexto do ambiente dos experimentos quanto os dados obtidos.

#### 3.1 Criação do Dataset

Foram utilizadas três placas de desenvolvimento com *chipset* ESP32 nos experimentos: o modelo *Wi-Fi ESP32 Bluetooth*, detalhado em Mohanan (2022) (possui 30 pinos e foi usado para testes de conexão Wi-Fi); o *ESP32 WROVER module*, detalhado em Najad (2019) (possui 38 pinos e foi usado para testes de conexão *General Packet Radio*

*Service* (GPRS)); e o *ESP32 LoRa V2 Heltec*, detalhado em Gunar Schorcht (possui 36 pinos e foi usado para testes de conexão LoRa). Exemplos de dispositivos usados estão ilustrados na Figura 1.

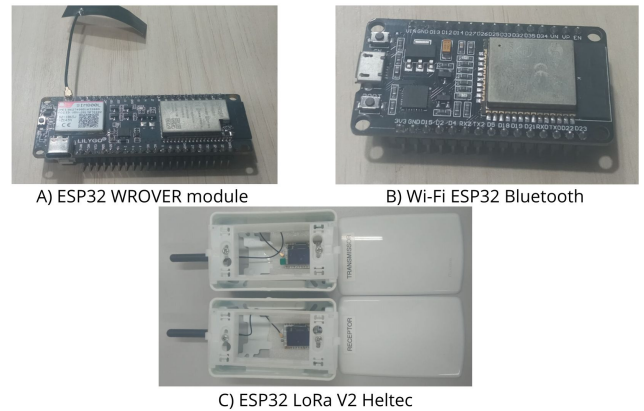


Figura 1. Exemplos de dispositivos usados para coleta dos dados.

Para cada placa, foi desenvolvido um código na linguagem de programação C para avaliar a distância, força de sinal, tempo de envio de um contador do transmissor para o receptor e o consumo energético. Essas métricas foram armazenadas para criar um *dataset* com os dados coletados. Uma vez que os dados de consumo energético foram contínuos e, em dispositivos IoT, a hibernação é uma operação crucial para economia de energia, uma função matemática (2) foi utilizada para estimar o consumo total, levando em conta a hibernação e o consumo contínuo.

Com os dados de consumo contínuo e a fórmula para estimar o consumo durante a hibernação, o *dataset* foi composto pelas seguintes colunas: tempo de hibernação (*deepsleep*); tipo de conexão (Wi-Fi, GPRS ou LoRa); consumo contínuo em mAh; tempo de conexão; tipo de bateria; capacidade da bateria em mAh; quantidade de envios que podem ser realizados pelo dispositivo por hora de forma contínua e com hibernação; e a duração do dispositivo (tempo que o dispositivo leva para ligar e enviar o pacote com sucesso) de forma contínua e com hibernação.

Os dados foram organizados em uma planilha e exportados para um arquivo CSV para serem processados por bibliotecas da linguagem de programação *Python*. O pré-processamento básico incluiu a conversão de dados numéricos com vírgula para ponto e a mudança do tipo de dado de *string* para *float*.

#### 3.2 Estimativa de Consumo Energético com Apoio de Técnicas de IA

O arranjo experimental montado leva em consideração cenários práticos relacionados com regiões que não possuem fácil acesso a infraestruturas de comunicação de qualidade, a exemplo de áreas rurais e aplicações, como Aquicultura 4.0. Nesse cenário, a eficiência energética desempenha um papel fundamental. Portanto, o objetivo principal é estimar a duração das baterias das placas, levando em consideração o tempo de funcionamento, o tempo de hibernação e o tipo de bateria.

Para iniciar a construção do sistema baseado em IA, o primeiro passo foi a aquisição de dados. Foi coletado um conjunto de dados contendo 1.250 entradas distribuídas em 10 colunas. Esses dados foram obtidos com base nos resultados anteriores do estudo, com a variável “*deepsleep*” variando de 1 a 60 e diferentes tipos de baterias: LR6, FR6, KR6 e ZR6, com capacidades médias de 2200, 3050, 800 e 1650 mAh, respectivamente.

Uma vez obtidos os dados, procedeu-se com uma análise da correlação entre as colunas. Para isso, foi realizado o cálculo de uma matriz de correlação de Pearson, que conforme Peng et al. (2022), é um indicador que varia de -1 a +1, em que 1 indica uma correlação positiva perfeita, -1 indica uma correlação inversa perfeita (valores próximos de 0 indicam uma ausência de correlação). A representação gráfica da matriz, aplicada aos dados deste estudo está disponível na Figura 2.

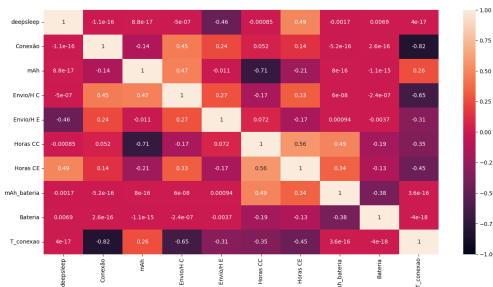


Figura 2. Matriz de correlação dos dados coletados no estudo.

Após observar que não havia colunas de dados com correlação total e que as correlações entre elas geralmente eram muito baixas, foram selecionadas as colunas que seriam usadas como variáveis-alvo. As variáveis escolhidas foram as seguintes: “Horas CC” (Horas de Consumo Contínuo), que representa o número de horas em que a placa opera continuamente; “Horas CE” (Horas de Consumo com Hibernação), que representa o número de horas em que a placa opera com o uso do método de hibernação; “Envios HC” (Envios Contínuos por Hora), que representa o número de envios que a placa realiza por hora continuamente; e “Envios HE” (Envios com Hibernação por Hora), que representa o número de envios que a placa realiza por hora respeitando a hibernação. Com essas colunas-alvo identificadas, foi realizada a escolha das colunas que seriam usadas como características para prever os resultados desejados.

Inicialmente, para cada coluna-alvo, foram consideradas todas as outras colunas como variáveis preditoras. No entanto, com o objetivo de simplificar o modelo e reduzir a quantidade de parâmetros necessários para obter respostas com alta precisão, optou-se por utilizar todas as colunas, exceto as quatro colunas-alvo.

Foram consideradas as seguintes técnicas de IA para geração dos modelos preditivos: *Linear Regression*, *Multilayer Perceptron*, *Decision Tree*, *Random Forest* e *K-Nearest Neighbors*. Nas LR e DT, foram mantidos os parâmetros padrão disponíveis na biblioteca, enquanto nas MLP, RF e KNN, realizou-se um ajuste paramétrico empírico. Para o KNN, foi utilizado  $k = 7$ . Na RF, a profundidade máxima

adotada foi 5. Já para MLP, foi incluído o “*early\_stopping*”, que interrompe o treinamento antes de atingir o número máximo de iterações, com o objetivo de evitar o *overfitting*. Além disso, definiu-se o número máximo de iterações como 1000 e foram utilizadas três camadas internas, cada uma com 100 neurônios.

Para avaliar os modelos de forma mais precisa, foi utilizado o método de validação cruzada com dez partições e cinco repetições, garantindo que todos os dados fossem cobertos. Após cada divisão dos dados em treinamento e teste, foi calculado o Erro Médio Quadrático (MSE), Raiz do Erro Quadrático Médio (RMSE), Erro Médio Absoluto (MAE), coeficiente de determinação (R2-score) e tempo gasto por cada algoritmo. Por fim, comparou-se a média dos resultados obtidos para as métricas de avaliação entre os modelos mencionados anteriormente.

## 4. RESULTADOS

Nesta seção, serão apresentados os resultados da pesquisa, abordando quais foram os modelos que obtiveram os melhores resultados na predição do consumo energético dos dispositivos.

### 4.1 Visão geral dos dados coletados

Como mencionado anteriormente, o consumo medido durante a coleta foi contínuo. Em outras palavras, se a placa não estiver no modo de hibernação, ela consumirá uma determinada quantidade de  $mA$  por hora. No entanto, ao considerar o modo de hibernação ativado, a placa ESP32 consumirá cerca de  $10 \mu A$  (microamperes). Considerando que o consumo da placa pode variar ao longo do tempo, dependendo se está ativa ou em hibernação, não se pode considerar apenas o consumo contínuo ao tentar prever a duração de uso da placa com uma bateria. Por isso, é importante calcular o consumo médio.

Para calcular o tempo de uso, divide-se o consumo da placa em dois tipos: consumo ativo e consumo em hibernação. Para calcular o consumo médio (CM) em microamperes, é utilizada a Equação 2, em que C representa o consumo em que o dispositivo opera em determinado modo em amperes, e P o intervalo de tempo em que o dispositivo opera em determinado modo em segundos.

$$C_m = \frac{C_1 \cdot P_1 + C_2 \cdot P_2 + \dots + C_n \cdot P_n}{P_1 + P_2 + \dots + P_n} \quad (2)$$

Além disso, realizou-se uma comparação entre a previsão do tempo de duração da bateria considerando o consumo contínuo e o consumo com hibernação. Essa comparação levou em consideração alguns dos principais tipos de pilhas AA utilizados, bem como suas capacidades em mAh, conforme ilustrado na Tabela 1.

Na Tabela 1, é possível observar a diferença de tempo em horas entre o consumo contínuo e o consumo com hibernação de 10 segundos. Essa diferença aumenta ainda mais conforme o tempo de hibernação é prolongado, pois, quanto maior o tempo de hibernação, menor será o tempo de atividade da placa e maior será a duração da bateria. O aumento proporcional do tempo pode ser visualizado de forma mais clara na Tabela 2.

Tabela 1. Comparativo de consumo e tempo de vida das baterias.

Consumo de Dispositivos em Operação Contínua (mAh)	Horas de Consumo Contínuo				Horas de Consumo com Hibernação			
	LR6	FR6	KR6	ZR6	LR6	FR6	KR6	ZR6
40	56	76	20	41	207	287	75	155
54	41	58	15	31	154	213	56	115
90	24	34	9	18	92	128	34	69

Tabela 2. Comparativo de horas de consumo e tempo de hibernação.

Baterias com Consumo Contínuo de 40 mAh	Duração (em Horas) do Dispositivo em Operação Contínua	Duração (em Horas) do Dispositivo em Operação com Hibernação			
		10 segundos	1 minuto	10 minutos	1 hora
LR6	56	207	965	8829	43946
FR6	76	287	1338	12240	60925
KR6	20	75	350	3210	15980
ZR6	41	155	724	6622	32959

Na Tabela 2, fica evidente a diferença no tempo de uso da placa entre o consumo contínuo e o consumo com hibernação, considerando diferentes tempos de hibernação. O aumento do tempo de hibernação resulta em um significativo aumento na vida útil da bateria, tornando a otimização do consumo de energia uma consideração crucial em cenários como os da Aquicultura 4.0.

#### 4.2 Avaliação dos modelos gerados por diferentes algoritmos de predição

Os modelos de predição obtidos por diferentes algoritmos foram comparados quando submetidos à predição das seguintes métricas: Horas de Consumo Contínuo (Horas CC), Horas de Consumo com Hibernação (Horas CE), Envios Contínuos por Hora (Envios HC) e Envios com Hibernação por Hora (Envios HE).

A partir da análise da Tabela 3, é possível perceber alguns pontos cruciais dos resultados obtidos. Por exemplo, é notável que, para métricas relacionadas a operações contínuas, como Envios Contínuos por Hora (Envios HC) e a quantidade de Horas de Consumo Contínuo (Horas CC), o coeficiente de determinação da maioria dos modelos é quase 100%, com erro praticamente nulo. No entanto, ao prever métricas com hibernação, como Envios com Hibernação por Hora (Envios HE) e a quantidade de Horas de Consumo com Hibernação (Horas CE), os resultados tendem a variar mais, mostrando que os dados para essas métricas são mais sensíveis e que há indícios que sugerem que os dados podem ter uma relação não-linear, exigindo uma comparação mais criteriosa dos modelos para tomada de decisão.

Ao tentar prever a quantidade de envios possíveis por hora com hibernação, devido à possível natureza não-linear dos dados, o modelo de Regressão Linear (LR) não obteve bom desempenho, com apenas 32,94% de coeficiente de determinação. Os modelos que se destacaram foram a Árvore de Decisão (DT) e a Floresta Aleatória (RF), com 95,96% e 97,62% de coeficiente de determinação, respectivamente. Embora o coeficiente de determinação do RF seja mais alta que a da DT, observa-se que a DT possui erros mais baixos que o RF. Além disso, a DT possui um tempo de execução consideravelmente mais rápido que o RF, sendo mais de 10 vezes mais rápido para executar.

Ao tentar prever a quantidade de horas que o dispositivo pode alcançar com hibernação (Horas CE), observa-se que o modelo DT supera os outros modelos, mantendo um

alto coeficiente de determinação (acima de 97%), baixo MSE, RMSE e MAE, e um tempo de execução baixo (0,82 segundos).

## 5. CONCLUSÕES

Esta pesquisa teve como principal objetivo propor mecanismos inteligentes de estimação de consumo energético de baterias de dispositivos para infraestruturas de comunicação. Este estudo é especialmente útil para apoiar a tomada de decisão em aplicações relacionadas com regiões com Internet de baixa qualidade, como em áreas rurais e, em especial, na Aquicultura 4.0. Neste estudo, foi conduzida uma análise do consumo de energia com base em diferentes tipos de baterias AA e diferentes períodos de hibernação. Os resultados demonstraram que a escolha adequada do período de hibernação pode aumentar significativamente a vida útil da bateria. Além disso, dados experimentais foram coletados de dispositivos reais e os dados foram usados para o treinamento de algoritmos de ML com foco em estimar o tempo de vida das baterias e o número de transmissões. Foram considerados os algoritmos *Linear Regression*, *Multi-Layer Perceptron*, *Decision Tree*, *Random Forest* e *K-Nearest Neighbors*. As métricas usadas para a comparação dos modelos foram o Erro Médio Quadrático (MSE), Raiz do Erro Quadrático Médio (RMSE), Erro Médio Absoluto (MAE), coeficiente de determinação (R2-score) e tempo de execução. Como os experimentos mostraram indícios da natureza não-linear dos dados, uma vez que a técnica de Regressão Linear não obteve bom desempenho, ao contrário de modelos não-lineares como Árvore de Decisão e *Random Forest*. O modelo de Árvore de Decisão obteve o melhor desempenho, com uma taxa de correlação média de 96,57% para a estimativa das Horas de Consumo com Hibernação (Horas CE) e o número de Envios com Hibernação por Hora (Envios HE). A Árvore de Decisão também apresentou bons resultados nas métricas MSE, RMSE, MAE e tempo de execução, com média superior aos demais modelos.

Como sugestões de trabalhos futuros, destacam-se: expandir o conjunto de dados de forma quantitativa e considerando outros dispositivos; e explorar técnicas relacionadas com aprendizagem profunda.

## AGRADECIMENTOS

Os autores agradecem o apoio financeiro fornecido pela Fundação de Amparo a Ciência e Tecnologia de Pernam-

Tabela 3. Avaliação dos modelos obtidos pelos diferentes algoritmos.

Modelos	MSE	RMSE	MAE	R2-score	Tempo (s)	Coluna-Alvo
LR	0,0025	0,0503	0,0248	0,9999	1,4834	Envios HC
MLP	922,8357	28,2185	16,6773	0,9998	209,6453	Envios HC
DT	0,0002	0,0100	0,0011	0,9999	0,9573	Envios HC
RF	0,0013	0,0358	0,0121	0,9999	16,7365	Envios HC
KNN	870,5834	11,9946	1,4434	0,9998	1,0985	Envios HC
LR	30422,2436	162,3333	84,3859	0,3294	1,2371	Envios HE
MLP	13285,5873	104,8465	43,1027	0,7059	82,2648	Envios HE
DT	1034,2026	13,6657	1,4335	0,9596	1,1412	Envios HE
RF	1367,6622	29,9598	10,1785	0,976254	14,6213	Envios HE
KNN	8298,9483	67,2946	12,0985	0,8875	0,9998	Envios HE
LR	166,3776	12,8566	8,6953	0,8925	1,3333	Horas CC
MLP	0,3988	0,5644	0,2732	0,9997	112,4233	Horas CC
DT	1,1615	3,3005	1,6132	1,0000	0,8997	Horas CC
RF	7,4264	2,7133	1,4085	0,9950	15,0166	Horas CC
KNN	1,2362	0,8268	0,1115	0,9991	0,8234	Horas CC
LR	232353,5008	473,7932	327,8656	0,7647	1,0933	Horas CE
MLP	61709,0535	213,9888	99,6435	0,9451	98,7988	Horas CE
DT	34002,9000	141,0554	38,4368	0,9718	0,8366	Horas CE
RF	59783,6657	228,3475	115,0788	0,9445	13,3765	Horas CE
KNN	64855,3734	199,3576	36,3728	0,9457	0,8220	Horas CE

buco - (FACEPE) e Universidade Federal Rural de Pernambuco - (UFRPE).

#### REFERÊNCIAS

- Ahsan, M., Based, M.A., Haider, J., and Rodrigues, E.M. (2021). Smart monitoring and controlling of appliances using lora based iot system. *Designs*, 5(1), 17.
- Bhattacharya, S., Maddikunta, P.K.R., Meenakshisundaram, I., Gadekallu, T.R., Sharma, S., Alkahtani, M., et al. (2021). Deep neural networks based approach for battery life prediction. *Computers, Materials & Continua*, 69(2).
- Callebaut, G., Leenders, G., Van Mulders, J., Ottoy, G., De Strycker, L., and Van der Perre, L. (2021). The art of designing remote iot devices—technologies and strategies for a long battery life. *Sensors*, 21(3), 913.
- Charbuty, B. and Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01), 20–28.
- Ergun, K., Ayoub, R., Mercati, P., and Rosing, T. (2019). Dynamic optimization of battery health in iot networks. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*, 648–655. IEEE.
- Faye, I., Fam, P.A., and Ndiaye, M.L. (2022). Energy consumption of iot devices: An accurate evaluation to better predict battery lifetime. *Radio Science*, 57(12), 1–10.
- Gunar Schorcht (????). Heltec WiFi LoRa 32 V2. [https://api.riot-os.org/group\\_boards\\_esp32\\_heltec-lora32-v2.html](https://api.riot-os.org/group_boards_esp32_heltec-lora32-v2.html). Accessed: 2024-4-2.
- Kohli, S., Godwin, G.T., and Urolagin, S. (2020). Sales prediction using linear and knn regression. In *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*, 321–329. Springer.
- Kraemer, F.A., Ammar, D., Braten, A.E., Tamkittikhun, N., and Palma, D. (2017). Solar energy prediction for constrained iot nodes based on public weather forecasts. In *Proceedings of the Seventh International Conference on the Internet of Things*, 1–8.
- Laghari, A.A., Wu, K., Laghari, R.A., Ali, M., and Khan, A.A. (2021). A review and state of art of internet of things (iot). *Archives of Computational Methods in Engineering*, 1–19.
- Liu, B., Zhao, Q., Jin, Y., Shen, J., and Li, C. (2021). Application of combined model of stepwise regression analysis and artificial neural network in data calibration of miniature air quality detector. *Scientific reports*, 11(1), 3247.
- Mohanan, V. (2022). DOIT ESP32 DevKit V1 WiFi development board - pinout diagram & arduino reference. <https://www.circuitstate.com/pinouts/doit-esp32-devkit-v1-wifi-development-board-\pinout-diagram-and-reference/>. Accessed: 2024-4-2.
- Najad (2019). ESP32 with onboard SIM800L ! <https://diyusthad.com/2019/07/esp32-with-onboard-sim800.html>. Accessed: 2024-4-2.
- Naskath, J., Sivakamasundari, G., and Begum, A.A.S. (2023). A study on different deep learning algorithms used in deep neural nets: Mlp som and dbn. *Wireless personal communications*, 128(4), 2913–2936.
- Peng, S., Han, W., and Jia, G. (2022). Pearson correlation and transfer entropy in the chinese stock market with time delay. *Data Science and Management*, 5(3), 117–123.
- Reddy Maddikunta, P.K., Srivastava, G., Reddy Gadekallu, T., Deepa, N., and Boopathy, P. (2020). Predictive model for battery life in iot networks. *IET Intelligent Transport Systems*, 14(11), 1388–1395.
- Schonlau, M. and Zou, R.Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, 20(1), 3–29.