UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Sistema de Análise Preditiva em Tempo Real para *Smart Meters* usando Machine Learning embarcado (TinyML).

JONES MÁRCIO NAMBUNDO

Itajubá, 18 de Julho de 2025

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JONES MÁRCIO NAMBUNDO

Sistema de Análise Preditiva em Tempo Real para *Smart Meters* usando Machine Learning embarcado (TinyML).

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração: Microeletrônica

Orientador: Prof. Dr. Otávio de Souza Martins Gomes

18 de Julho de 2025

Itajubá

Agradecimentos

Primeiramente, agradeço a Deus por me conceder força, sabedoria e por me mostrar os caminhos certos para não desistir no meio da caminhada. À minha família de forma geral, e em especial à minha mãe, Adelaide Nambundo, e ao meu pai, Sanda, por todo o suporte e conselhos. À minha esposa, aos meus filhos e sobrinhos, pelo amor incondicional vocês foram meu alicerce em cada momento, proporcionando apoio e encorajamento, mesmo nos tempos mais desafiadores.

Ao meu orientador, Otávio de Souza Martins Gomes, deixo meu profundo agradecimento pela orientação, paciência, dedicação e pelos desafios propostos. Suas palavras de incentivo e conselhos foram essenciais para o meu crescimento acadêmico e pessoal. Sua sabedoria e experiência me guiaram em cada passo, ajudando-me a superar obstáculos e alcançar novos horizontes.

Aos meus colegas e amigos, Sílvio Sebastião, Herman Jaime, Hermane Bráulio, Fidel Adanvo, Daniel Quiteque e Milton Semedo, Giorgino Baltazar, com quem compartilhei inúmeras horas de estudo, discussões e aprendizados. Cada um de vocês contribuiu de forma única para a minha jornada, e sou grato por termos percorrido esse caminho juntos. As amizades que construímos ao longo desses anos são um tesouro que levarei comigo para sempre.

Agradeço também aos demais professores que, ao longo dessa trajetória, partilharam seu conhecimento e paixão pelo ensino. Sua dedicação e compromisso com a educação foram inspirações constantes.

Gostaria de expressar meu sincero agradecimento à Universidade Federal de Itajubá (UNIFEI) pela bolsa de estudos concedida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro e pela valiosa parceria ao longo deste projeto.

Por fim, a todos que, direta ou indiretamente, contribuíram para que eu alcançasse este objetivo: suas palavras de apoio, gestos de carinho e incentivos fizeram toda a diferença. Este trabalho não seria possível sem a colaboração de cada um de vocês. Que esta conquista seja apenas o começo de novas jornadas e desafios. A todos, o meu mais sincero muito obrigado.

"Tudo me é permitido, mas nem tudo convém; tudo me é permitido, mas eu não deixarei que nada me domine." (1 Coríntios 6:12)

Resumo

O crescente foco global na eficiência energética impulsiona a necessidade de soluções inovadoras que combinem sustentabilidade, economia e tecnologia. Nesse contexto, os medidores inteligentes *(smart meters)* desempenham um papel essencial ao permitir o monitoramento em tempo real do consumo de energia, promovendo transparência para provedores e consumidores. Este trabalho apresenta o desenvolvimento de um sistema de análise preditiva em tempo real para *smart meters*, utilizando *Machine Learning* embarcado (*TinyML*) no microcontrolador ESP32. O sistema é projetado para operar em ambientes com conectividade limitada, realizando processamento local dos dados e reduzindo a dependência de infraestrutura em nuvem.

O protótipo utiliza dados reais coletados de uma geladeira durante 31 dias, com medições contínuas 24 horas por dia. Três modelos foram testados e dois implementados: *XGBoost regressor* para previsão de consumo, e *One-Class SVM* e *Autoencoder* para detecção de anomalias. Os modelos foram otimizados para execução embarcada: o *One-Class SVM* e o *XGBoost* foram convertidos para C++ usando a biblioteca *micromlgen*, enquanto o *Autoencoder* foi adaptado para *TensorFlow Lite* (*TFLite*) com técnicas de *pruning* para reduzir seu tamanho e consumo computacional.

Os resultados mostraram que o *One-class SVM* alcançou maior precisão na detecção de anomalias, enquanto o *Autoenconder* apresentou menor tempo de inferência e menor uso de memória, tornando-o uma solução eficiente para dispositivos embarcados. O *XGBoost* demonstrou um MAE de 0.27 na previsão do consumo, indicando um bom desempenho na acurácia das suas previsões. Esses resultados destacam a viabilidade da aplicação de *TinyML* em sistemas de monitoramento de energia, contribuindo para a gestão eficiente das redes de distribuição, detecção de fraudes e promoção de comportamentos mais sustentáveis entre os usuários.

Palavras-chaves: *TinyML*, *Smart Meters*, Análise Preditiva em Tempo Real, Eficiência Energética, *Machine Learning*, *Big Data*

Abstract

The growing global focus on energy efficiency drives the need for innovative solutions that combine sustainability, economy, and technology. In this context, smart meters play an essential role by enabling real-time monitoring of energy consumption, promoting transparency for both providers and consumers. This work presents the development of a real-time predictive analysis system for smart meters, utilizing embedded Machine Learning (TinyML) on the ESP32 microcontroller. The system is designed to operate in environments with limited connectivity, performing local data processing and reducing reliance on cloud infrastructure.

The prototype uses real data collected from a refrigerator over 31 days, with continuous 24-hour measurements. Three models were tested and two implemented: XGBoost regressor for consumption forecasting, and One-Class SVM and Autoencoder for anomaly detection. The models were optimized for embedded execution: the One-Class SVM and XGBoost were converted to C++ using the micromlgen library, while the Autoencoder was adapted for TensorFlow Lite (TFLite) with pruning techniques to reduce its size and computational consumption.

The results showed that the One-Class SVM achieved higher accuracy in anomaly detection, while the Autoencoder presented lower inference time and memory usage, making it an efficient solution for embedded devices. The XGBoost demonstrated a Mean Absolute Error (MAE) of 0.27 in consumption forecasting, indicating good performance in the accuracy of its predictions. These results highlight the viability of applying TinyML in energy monitoring systems, contributing to the efficient management of distribution networks, fraud detection, and the promotion of more sustainable behaviors among users

Key-words: TinyML, Smart Meters, Real-time Predictive Analysis, Energy Efficiency, Machine Learning, Big Data

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

1D-CNN	*One-Dimensional Convolutional Neural Network*	31
AC	*Alternating Current*	55
ADC	*Analog-to-Digital Converter*	62
AE	*Autoencoder*	37
AMI	*Advanced Metering Infrastructure*	19
B2B	*Business-to-Business*	31
B2C	*Business-to-Consumer*	31
CNN	*Convolutional Neural Network*	41
CT	*Current Transformer*	54
DC	*Direct Current*	55
DL	*Deep Learning*	33
DNN	*Deep Neural Network*	32
DQN	*Deep Q-Networks*	30
FN	*False Negative*	70
FP	*False Positive*	70
GBM	*Gradient Boosting Machines*	36
ICT	*Information and Communication Technologies*	20
IEA	*International Energy Agency*	19
IOT	*Internet of Things*	17
LCD	*Liquid Crystal Display*	55
LED	*Light Emitting Diode*	55
LGPD	*Lei Geral de Proteção de Dados Pessoais*	16
LPWAN	*Low-Power Wide Area Network*	25
LSTM	*Long Short-Term Memory*	31
ML	*Machine Learning*	15
MQTT	*Message Queuing Telemetry Transport*	60
MSE	*Mean Squared Error*	38
NIST	*National Institute of Standards and Technology*	20
OTA	*Over-The-Air*	60
PCA	*Principal Component Analysis*	29
PCB	*Printed Circuit Board*	58
PLC	*Power Line Communication*	25
RC	*Resistor-Capacitor*	57
RL	*Reinforcement Learning*	30
RMS	*Root Mean Square*	24

SG	*Smart Grids*	15
SGAM	*Smart Grid Architecture Model*	20
SM	*Smart Meters*	15
SVM	*Support Vector Machine*	17
TFLite	*TensorFlow Lite*	17
TN	*True Negative*	70
TP	*True Positive*	70
VDC	*Volts Direct Current*	56
Wi-Fi	*Wireless Fidelity*	25
XGBoost	*Extreme Gradient Boosting*	17
Zigbee	*Zigbee Communication Protocol*	25

Lista de símbolos

E	*Energia total consumida em kWh (Total Energy Consumed)*	13
I	*Corrente elétrica (Electric Current)*	47
P	*Potência ativa (Active Power)*	47
Q	*Potência reativa (Reactive Power, VAR)*	47
R	*Resistência elétrica (Electrical Resistance, Ω)*	48
S	*Potência aparente (Apparent Power, VA)*	47
V	*Tensão (Voltage)*	47
V_{RMS}	*Tensão eficaz (RMS Voltage)*	49
V_{pico}	*Tensão de pico (Peak Voltage)*	49
$\sin(\theta)$	*Seno do ângulo de fase entre corrente e tensão (Sine of Phase Angle)*	47
CO_2	*Dióxido de carbono (Carbon Dioxide)*	19
$i(t)$	*Valor da corrente em amperes no instante t (Current, A)*	46
$i[n]$	*Valor da corrente em amperes na amostra n (Current sample, A)*	46
$p(t)$	*Valor da potência instantânea em watts no instante t (Instantaneous Power, W)*	46
$p[n]$	*Valor da potência instantânea em watts na amostra n (Instantaneous Power sample, W)*	46
t	*Tempo de operação (Operation Time)*	48
$v(t)$	*Valor da tensão em volts no instante t (Voltage, V)*	46
$v[n]$	*Valor da tensão em volts na amostra n (Voltage sample, V)*	46

Sumário

1 Introdução

O sistema energético brasileiro destaca-se por sua matriz predominantemente renovável, com 85% da energia proveniente de fontes como hidrelétricas, eólicas, biomassa e solar [1]. Essa configuração reflete uma trajetória histórica centrada em hidrelétricas, resultando em uma infraestrutura de baixo carbono. Nos últimos anos, a diversificação da geração de energia, impulsionada pela incorporação de fontes eólicas e solares, expandiu significativamente a segurança energética e reduziu impactos ambientais [1].

Essa transformação exige a modernização da infraestrutura elétrica, tornando as redes mais inteligentes e eficientes. A digitalização da rede elétrica e a adoção de tecnologias avançadas são fundamentais para otimizar a distribuição de energia e integrar novas fontes renováveis. Nesse contexto, os medidores inteligentes SM (*Smart Meters*) emergem como ferramentas estratégicas para aprimorar o consumo e proporcionar maior transparência operacional [2].Esses dispositivos permitem a coleta de dados em tempo real, capacitando consumidores a monitorar e ajustar seus hábitos de forma eficiente. Iniciativas como a Tarifa Branca incentivam o uso da eletricidade em horários de menor demanda, promovendo um equilíbrio mais eficiente na gestão da rede elétrica [2].

Embora a adoção dos medidores inteligentes no Brasil ainda esteja em fase inicial, projeções indicam um crescimento expressivo. Estima-se que o número de unidades instaladas aumente de 901 mil em 2022 para 3,9 milhões até 2030, com um crescimento médio anual de 20,1% [3]. Esse avanço é impulsionado pela expansão da geração distribuída, especialmente sistemas fotovoltaicos, já presentes em mais de 1,2 milhão de residências [3]. Além de contribuir para a estabilidade da rede, esses dispositivos possibilitam uma participação mais ativa dos consumidores na gestão energética.

A crescente demanda por energia reforça a necessidade de soluções tecnológicas que otimizem o consumo. Nesse cenário, a integração de técnicas de Machine Learning ML (*Machine Learning*) aos medidores inteligentes permite prever padrões de consumo e detectar anomalias, aumentando a confiabilidade do sistema e reduzindo perdas operacionais [4, 5]. Uma inovação relevante é o uso de TinyML, que habilita a execução de algoritmos de aprendizado de máquina diretamente em dispositivos de baixa potência, como o ESP32. Essa abordagem reduz custos operacionais, garante maior privacidade ao evitar transmissões contínuas para servidores centrais e se torna especialmente útil em regiões remotas, onde a conectividade é limitada [6].

As redes elétricas inteligentes SG (*Smart Grids*) desempenham um papel crucial na integração de fontes renováveis e na gestão dinâmica da oferta e demanda de energia. No entanto, desafios como segurança cibernética e proteção de dados exigem conformidade

com regulamentações como a LGPD (*Lei Geral de Proteção de Dados Pessoais*), garantindo a confiança dos usuários e a confiabilidade das medições inteligentes [1]. Assim, a combinação de medidores inteligentes e aprendizado de máquina representa uma oportunidade significativa para modernizar a infraestrutura energética brasileira, promovendo sustentabilidade, eficiência e maior controle sobre o consumo de energia. Neste contexto, esta dissertação propõe o desenvolvimento e a validação de um protótipo inovador para análise preditiva em tempo real em medidores inteligentes, empregando algoritmos de Machine Learning embarcado. O objetivo é criar um sistema que aprimore a resposta das redes elétricas, forneça previsões de consumo de alta precisão e promova práticas sustentáveis.

1.1 Contextualização do Problema

A gestão eficiente da energia é uma preocupação global devido à sua relevância econômica e ambiental. Investimentos em tecnologias inteligentes de monitoramento geraram economias expressivas em diversos países, evidenciando a necessidade de soluções inovadoras para otimizar o consumo energético [7]. Para concessionárias, prever a demanda e identificar fraudes em tempo real é fundamental para reduzir perdas não técnicas e melhorar a eficiência operacional [8]. Para consumidores, a análise preditiva permite entender padrões de consumo e adotar práticas mais conscientes, incentivando a economia de energia [9, 10].

Assim, técnicas de *Machine Learning* aplicadas a medidores inteligentes demonstram grande potencial para monitoramento e previsão do consumo energético. Algoritmos como Regressão Linear, Árvores de Decisão, Florestas Aleatórias e Redes Neurais Profundas possibilitam a análise de grandes volumes de dados, identificando padrões complexos e gerando previsões precisas. A combinação dessas abordagens, conhecida como *Ensemble Learning*, destaca-se por sua robustez e acurácia [11, 12].

Adicionalmente, a utilização de TinyML representa um avanço significativo ao permitir o processamento local de dados em dispositivos de baixa potência, como os próprios medidores inteligentes . Essa abordagem reduz latência, elimina a necessidade de transmissão contínua para servidores centrais e minimiza custos operacionais, sendo ideal para aplicações contínuas e de baixo impacto ambiental [6].

1.2 Justificativa

Diante das mudanças climáticas e do aumento da demanda por eletricidade, é essencial desenvolver soluções que otimizem o consumo energético. Embora os medidores inteligentes já forneçam dados detalhados, seu potencial pode ser ampliado com a aplica-

ção de *Machine Learning* embarcado. O uso de TinyML nesses dispositivos permite não apenas a detecção de anomalias e fraudes, mas também operações de baixo custo e alta eficiência energética. Além disso, o processamento local melhora a privacidade e segurança dos dados, reduzindo riscos associados à transmissão de informações sensíveis [6].

Essa abordagem beneficia tanto concessionárias quanto consumidores. Para empresas, a previsão de demanda e a detecção de irregularidades em tempo real reduzem perdas e melhoram a eficiência operacional. Para usuários, a análise preditiva oferece uma visão detalhada do consumo e incentiva práticas mais sustentáveis.

1.3 Objetivo

1.3.1 Objetivo Geral

Desenvolver um sistema embarcado de análise preditiva em tempo real utilizando TinyML para monitorar e prever o consumo de energia registrado por medidores inteligentes. O sistema busca otimizar a eficiência energética, identificar padrões de consumo, prever demandas futuras e detectar anomalias ou fraudes, contribuindo para a melhoria contínua dos serviços de energia.

1.3.2 Objetivos Específicos

- Desenvolver um protótipo de baixo custo e economicamente viável, utilizando técnicas de TinyML para processamento local de dados em dispositivos de IOT (*Internet of Things*), com o ESP32.
- Coletar e processar dados reais de consumo energético de uma geladeira durante 31 dias, operando continuamente por 24 horas, garantindo uma base sólida para treinamento e validação dos modelos.
- Implementar modelos de aprendizado de máquina voltados à previsão de consumo e detecção de anomalias, utilizando o XGBoost (*Extreme Gradient Boosting*) para estimativa do consumo de energia, e os modelos One-Class SVM (*Support Vector Machine*) e Autoencoder para identificação de outliers ou comportamentos atípicos.
- Otimizar e comparar os modelos no ESP32, convertendo one - Class SVM e XGBoost para C++ via micromlgen e Autoencoder para TFLite (*TensorFlow Lite*) com técnicas de pruning, avaliando métricas como precisão, uso de memória e tempo de inferência.

- Fornecer insights personalizados sobre o uso de energia, incluindo horários de pico e sugestões de otimização, apresentados em um painel web intuitivo e de fácil interpretação.

1.4 Estrutura da Dissertação

A dissertação está organizada da seguinte maneira:

O Capítulo 1 apresenta a introdução, incluindo o contexto do problema, a justificativa, os objetivos e a estrutura geral do trabalho.

O Capítulo 2 apresenta a revisão da literatura, que discute os principais conceitos sobre eficiência energética, medidores inteligentes, tecnologias de *big data*, *machine learning* e *TinyML*, além apresentar e analisar trabalhos relacionados.

O Capítulo 3 apresenta a fundamentação teórica, detalhando os modelos e técnicas de machine learning, a análise preditiva de consumo energético e os desafios no processamento de dados em tempo real com TinyML.

O Capítulo 4 descreve a metodologia adotada, incluindo os materiais utilizados, a arquitetura do sistema, o processo de coleta e tratamento dos dados, os procedimentos de treinamento e avaliação dos modelos, bem como a implementação das soluções TinyML no dispositivo embarcado.

O Capítulo 5 apresenta o desenvolvimento do sistema, explicando a implementação dos modelos de machine learning no ambiente TinyML, a integração dos sensores e dispositivos, os testes realizados, a comparação entre modelos, além das conclusões e sugestões para trabalhos futuros.

Os apêndices incluem materiais produzidos pelo autor, como as fórmulas utilizadas no Arduino IDE (Apêndice A) e o detalhamento dos custos do protótipo (Apêndice B). Os anexos apresentam os artigos científicos publicados durante o desenvolvimento da pesquisa, reforçando a contribuição acadêmica do trabalho.

2 Revisão Da Literatura

Este capítulo apresenta uma visão abrangente dos principais conceitos e tecnologias relacionados ao tema deste estudo. A revisão inclui a análise da eficiência energética e sua relevância global, a evolução das smart grids, as tecnologias de *Big Data* e *Machine Learning* (ML) aplicadas ao setor energético, além do conceito e aplicação de TinyML. A seção de Infraestrutura Avançada de Medição AMI (*Advanced Metering Infrastructure*) e Smart Meters explora como essas tecnologias se tornaram essenciais para redes inteligentes e a modernização do setor energético.

2.1 Eficiência Energética e sua Relevância Global

A eficiência energética é um pilar fundamental para a sustentabilidade global, com impacto direto na redução do consumo de recursos naturais e na mitigação das emissões de gases de efeito estufa. A crescente demanda por energia, impulsionada pelo desenvolvimento industrial, urbanização e crescimento populacional, exerce pressão sobre as reservas de combustíveis fósseis, que ainda representam a principal fonte energética em muitos países. Medidas voltadas à eficiência energética ajudam a prolongar a vida útil desses recursos, reduzir a dependência de fontes não renováveis e diminuir custos energéticos para consumidores residenciais, comerciais e industriais, promovendo um uso mais sustentável e acessível [13, 14].

No contexto global, a eficiência energética desempenha um papel estratégico nas políticas de mitigação das mudanças climáticas. Segundo a Agência Internacional de Energia IEA (*International Energy Agency*), a eficiência energética pode contribuir com até 40% da redução necessária das emissões de dióxido de carbono CO_2 para atingir as metas estabelecidas no Acordo de Paris [15]. Essa abordagem é considerada eficaz, pois permite a redução das emissões sem comprometer o desenvolvimento econômico. Setores como transporte, construção civil e indústria apresentam grande potencial para economias de energia, gerando empregos verdes e fortalecendo economias locais. Além de seus benefícios ambientais e econômicos, a eficiência energética também é crucial para a segurança energética. Muitos países dependentes da importação de combustíveis fósseis enfrentam desafios relacionados à volatilidade dos preços e à instabilidade política em regiões produtoras. Nesse cenário, a eficiência energética surge como uma solução estratégica para reduzir essa vulnerabilidade, melhorando a resiliência dos sistemas energéticos e diminuindo a necessidade de importações [15]. Além disso, a adoção de medidas eficientes contribui para estabilizar a rede elétrica e evitar apagões, especialmente em países com infraestrutura limitada [15].

A integração de fontes renováveis, como a energia solar e eólica, beneficia-se significativamente da adoção de práticas de eficiência energética. Essas fontes são intermitentes e dependem de condições naturais, como vento e insolação, o que pode gerar flutuações na produção de energia. Nesse contexto, a eficiência energética atua como um fator de otimização, permitindo ajustar o consumo à disponibilidade da geração e maximizando o aproveitamento das energias renováveis [16]. Tecnologias como medidores inteligentes e análises preditivas desempenham um papel essencial na gestão eficiente da distribuição de energia, contribuindo para o equilíbrio entre oferta e demanda [15, 16]. Além disso, a eficiência energética é um componente central das políticas de desenvolvimento sustentável. Diversos países têm implementado regulamentações e estabelecido padrões mínimos de eficiência para eletrodomésticos, veículos e edificações, com o objetivo de reduzir o impacto ambiental e impulsionar a transição para uma economia de baixo carbono. A colaboração entre setores público e privado é fundamental para alcançar os objetivos globais de desenvolvimento sustentável e garantir que o crescimento econômico ocorra sem comprometer a saúde do planeta. Nesse cenário, a iniciativa privada exerce um papel estratégico, investindo em tecnologias eficientes e adotando práticas sustentáveis [16].

2.2 Redes Elétricas Inteligentes (Smart Grids): Conceito e Arquitetura

As Smart Grids, ou redes elétricas inteligentes, são sistemas modernos que representam uma transformação significativa na forma como a energia elétrica é gerada, distribuída e consumida. O Instituto Americano de Tecnologia e Padrões NIST (*National Institute of Standards and Technology*) define Smart Grid como: *"uma rede moderna que permite o fluxo bidirecional de energia, usando comunicação nos dois sentidos e técnicas de controle, que possibilitará novas funcionalidades e novas aplicações"* [17].

Esses sistemas integram tecnologias de informação e comunicação ICT (*Information and Communication Technologies*) à infraestrutura elétrica tradicional, proporcionando maior eficiência, sustentabilidade e resiliência ao sistema elétrico [15]. Diferente das redes convencionais, caracterizadas por um fluxo unidirecional de energia e monitoramento limitado, as Smart Grids permitem o gerenciamento dinâmico da demanda, a integração de fontes renováveis e a automação de processos, favorecendo tanto consumidores quanto operadores do sistema. A arquitetura das Smart Grids baseia-se em múltiplas camadas de interoperabilidade, conectando componentes físicos, como medidores inteligentes e sensores, a uma infraestrutura de comunicação robusta e segura. Na Europa, o modelo SGAM (*Smart Grid Architecture Model*) é amplamente utilizado para descrever e garantir a comunicação eficaz entre os diferentes elementos dessas redes [18]. Como ilustrado na Tabela 3, as Smart Grids diferenciam-se das redes tradicionais em vários

aspectos, incluindo fluxo bidirecional de energia, monitoramento em tempo real, alta capacidade de integração de fontes renováveis e detecção instantânea de falhas. No Brasil, o sistema elétrico regulado pela ANEEL conta com a supervisão do Operador Nacional do Sistema Elétrico, responsável por gerenciar o Sistema Interligado Nacional [19].

Tabela 3 – Comparação entre Redes Tradicionais e Smart Grids

Aspecto	**Redes Tradicionais**	**Smart Grids**
Fluxo de Energia	Unidirecional	Bidirecional
Monitoramento	Manual e intermitente	Automático e em tempo real
Integração de Renováveis	Limitada	Alta
Detecção de Falhas	Demorada	Instantânea
Resposta à Demanda	Inexistente	Implementação ativa de preços dinâmicos
Segurança e Privacidade	Limitada	Avançada, com criptografia.
Eficiência Energética	Baixa	Alta, com automação
Confiabilidade	Suscetível a falhas	Alta, com redundância e recuperação rápida
Participação do Consumidor	Passiva	Ativa, com monitoramento remoto

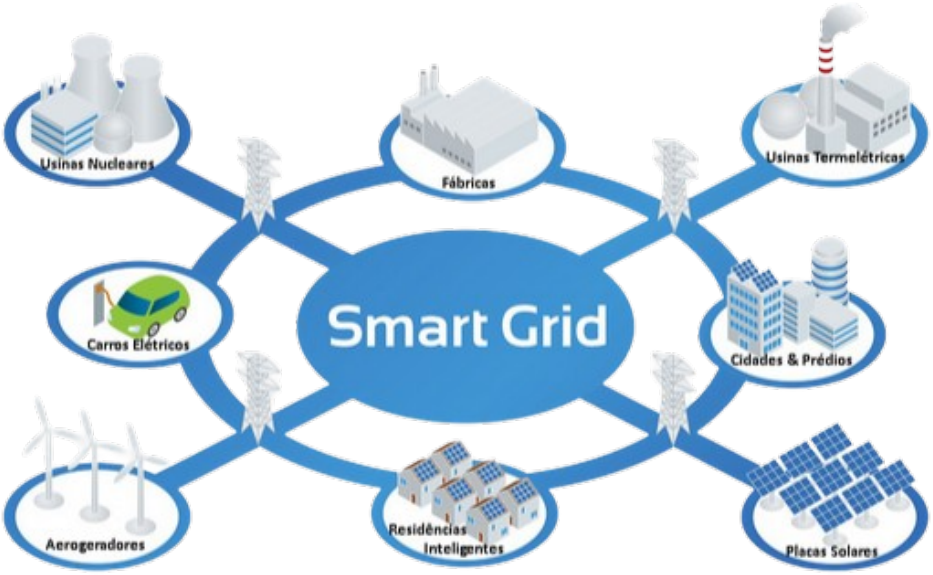
Fonte: Adptado de [19]

Uma das principais inovações associadas às redes elétricas inteligentes é o uso de equipamentos eletrônicos capazes de processar e comunicar informações em tempo real. Essa capacidade facilita o gerenciamento de dados e o controle operacional, automatizando processos de monitoramento e proteção, promovendo a segurança e confiabilidade do fornecimento de energia. O conceito de Smart Grids consolidou-se a partir do artigo seminal "Toward a Smart Grid", publicado por Amin e Wollenberg em 2005 [20], embora ideias precursoras tenham surgido ainda em 1998. Atualmente, concessionárias de energia estão focadas em três pilares principais para evoluir as redes elétricas: **(1) melhoria da infraestrutura física** [20], **(2) adição de uma camada digital de comunicação e processamento de dados** [21], e **(3) transformação dos processos de negócio para maximizar o uso das novas tecnologias** [**22**].

No entanto, a implementação ampla das Smart Grids enfrenta desafios significativos. A proteção contra ameaças cibernéticas, a adaptação a novos fluxos de informação e a necessidade de investimentos em infraestrutura são questões críticas para a estabilidade e segurança das redes modernas [23]. No Brasil, esses desafios são acentuados em regiões com menor desenvolvimento tecnológico. Apesar disso, as Smart Grids prometem reduzir perdas energéticas, aumentar a confiabilidade do sistema e facilitar a integração de fontes renováveis, tornando esses esforços altamente compensatórios.

Como ilustrado na figura 1, as Smart Grids promovem uma conexão dinâmica e eficiente entre produtores e consumidores de energia, viabilizando um sistema de rede elétrica mais flexível e adaptável. Esse modelo não apenas atende à crescente demanda energética, mas também incorpora práticas de gestão sustentável, configurando-se como um passo importante na transição para um futuro energético mais sustentável e seguro.

Figura 1 – Fluxo Dinâmico de Energia em Smart Grids



Fonte: Figura extraida de [24]

2.3 Infraestrutura Avançada de Medição

A *Advanced Metering Infrastructure* (AMI), ou Infraestrutura Avançada de Medição, desempenha um papel essencial nas redes elétricas inteligentes, viabilizando uma comunicação bidirecional entre consumidores e concessionárias. Constituída por medidores inteligentes, sensores e dispositivos de comunicação, a AMI coleta e transmite dados em tempo real sobre o consumo de energia, permitindo o monitoramento remoto e a gestão eficiente da demanda [25]. Essa tecnologia não apenas automatiza as leituras de consumo, mas também fornece informações precisas que incentivam um uso mais consciente da energia [26]. A operação da AMI depende de sistemas de comunicação robustos, garantindo a segurança e privacidade dos dados dos consumidores. Esses sistemas permitem que as concessionárias respondam rapidamente a mudanças na demanda e identifiquem problemas de fornecimento em tempo real, contribuindo para a confiabilidade da rede [27].

Conforme ilustrado na Figura 2, a AMI integra diversos componentes de medição e comunicação, promovendo uma operação sincronizada. Além disso, em combinação com sistemas de resposta à demanda, a AMI possibilita a implementação de tarifas dinâmicas, ajustando os preços conforme o consumo em horários de pico e favorecendo a eficiência

energética [28]. A AMI oferece vantagens econômicas tanto para consumidores quanto para concessionárias. A automação da coleta de dados elimina a necessidade de leituras manuais, reduzindo custos operacionais. Além disso, os consumidores podem ajustar seus hábitos de consumo com base em informações em tempo real, economizando energia e reduzindo suas contas [29].

Figura 2 – Infraestrutura de medição avançada

Fonte: Figura extraída de [28]

Apesar de seus benefícios, a implementação da AMI enfrenta desafios, como a necessidade de infraestrutura de comunicação confiável e a adaptação a diferentes padrões de medição em regiões diversas. Avanços tecnológicos estão sendo desenvolvidos para superar essas barreiras, possibilitando a expansão da AMI em larga escala [30]. A Infraestrutura Avançada de Medição é uma peça-chave para o sucesso das redes elétricas inteligentes, promovendo eficiência, sustentabilidade e inovação no setor energético.

2.4 Medidores inteligentes: Conceitos e Evolução

Os medidores inteligentes, ilustrados na Figura 3, são dispositivos digitais que medem e registram o consumo de eletricidade, gás ou água em tempo real, transmitindo essas informações para as concessionárias de serviços públicos. Eles desempenham um papel essencial nas Smart Grids, substituindo os medidores tradicionais e oferecendo funcionalidades avançadas, como leitura remota e controle automatizado [31, 32].

A comunicação bidirecional proporcionada pelos smart meters permite que consumidores e concessionárias ajustem o consumo com base em dados em tempo real, possibilitando sistemas de precificação dinâmica, onde a eletricidade se torna mais barata em períodos de baixa demanda. Assim, esses dispositivos contribuem para a redução de custos e promovem o uso eficiente dos recursos energéticos.

Com o avanço tecnológico, os medidores inteligentes atendem à crescente demanda por eficiência e integração de fontes renováveis, preparando a rede elétrica para desafios futuros. A capacidade de monitorar e controlar cargas remotamente aumenta a segurança

e facilita o gerenciamento tanto para concessionárias quanto para consumidores [15]. Diferente da medição analógica, que exige leituras manuais, a medição inteligente fornece dados precisos e atualizados, permitindo uma gestão mais eficiente do uso de energia. Dada a crescente preocupação com a sustentabilidade, os smart meters se apresentam como uma solução inovadora para residências, pequenas empresas e comércios. Além de medir e registrar o consumo em tempo real, os medidores inteligentes possibilitam a captura detalhada de informações relacionadas à qualidade do sinal elétrico, tais como a composição harmônica, além de medições de corrente, tensão e potência. Esses dados são disponibilizados tanto em valores instantâneos quanto em valores eficazes, calculados pela raiz do valor quadrático médio RMS (*Root Mean Square*) [31].

O acesso a essas informações em tempo real abre caminho para a aplicação de técnicas avançadas de análise, como algoritmos de machine learning, métodos de otimização e modelos preditivos para monitoramento e gestão da rede elétrica [33]. Essas técnicas permitem detectar anomalias, prever demandas futuras e otimizar o uso dos recursos energéticos. Para viabilizar essas aplicações, é necessário que os dados coletados pelos medidores sejam transmitidos para sistemas centralizados via protocolos de comunicação IoT, armazenados em bancos de dados especializados para séries temporais, e pré-processados para extração de características relevantes. Posteriormente, esses dados alimentam modelos computacionais capazes de gerar insights valiosos para concessionárias e consumidores, contribuindo para uma rede mais eficiente, segura e sustentável. [14]

Figura 3 – Smart Meters Energy

Fonte: Figura extraída de [34]

Nos últimos dez anos, a instalação desses dispositivos triplicou, e a expectativa é de que 93% dos sistemas de medição adotem essa tecnologia nos próximos anos. No final de 2023, a quantidade de medidores inteligentes instalados em países-chave reforça essa tendência: no Reino Unido, aproximadamente 29,5 milhões de medidores estavam em operação, de acordo com o UK *Government Publishing Service* [35]. Nos Estados Unidos, o número foi ainda maior, com cerca de 128 milhões de dispositivos instalados [36], na China, o número atingiu cerca de 496 milhões de medidores até o final de 2023. Esses números refletem a rápida adoção dos medidores inteligentes ao redor do mundo, demonstrando sua importância na gestão eficiente de energia e recursos. Os smart meters oferecem uma série de benefícios que os tornam essenciais para a modernização das redes de serviços públicos, conforme detalhado em [37]:

- Eficiência energética: permitem que consumidores monitorem seu consumo em tempo real, incentivando a redução do uso em horários de pico.
- Gerenciamento da rede: as concessionárias podem detectar falhas, como quedas de energia, com mais precisão e agilidade.
- Automação de processos: elimina a necessidade de leituras manuais, reduzindo custos operacionais e erros humanos.
- Dados em tempo real: a análise de dados permite melhorar o planejamento de infraestrutura e políticas de fornecimento.

O funcionamento dos medidores inteligentes envolve uma rede de comunicação que coleta e transmite dados de consumo para um sistema central das concessionárias. Entre as tecnologias de comunicação utilizadas, destacam-se Zigbee (*Zigbee Communication Protocol*), Wi-Fi (*Wireless Fidelity*), PLC (*Power Line Communication*) e LPWAN (*Low-Power Wide Area Network*), cada uma com características próprias para atender a diferentes necessidades de conectividade.

O Zigbee é uma solução sem fio eficiente e de baixo custo, operando na frequência de 2,4 GHz, ideal para comunicação entre medidores inteligentes e concentradores, especialmente em áreas urbanas densas [38]. O Wi-Fi, por sua alta velocidade e ampla cobertura, é comumente utilizado em redes domésticas e também se aplica a sistemas de medição inteligente devido à sua capacidade de suportar múltiplos dispositivos simultaneamente [39]. O PLC, por sua vez, utiliza a infraestrutura da rede elétrica existente para transmitir dados, sendo uma opção vantajosa em locais com conectividade sem fio limitada ou inviável [40]. Complementando essas tecnologias, as redes LPWAN destacam-se por conectar dispositivos de baixo consumo de energia em longas distâncias. Projetadas para aplicações de IoT, como medidores inteligentes e sensores agrícolas, as redes LPWAN são ideais para áreas remotas ou com infraestrutura de comunicação precária. Sua principal

vantagem é a capacidade de operar por longos períodos com baterias de longa duração. Exemplos de tecnologias LPWAN incluem LoRa, NB-IoT e Sigfox, cada uma adaptada a diferentes cenários de uso e requisitos de conectividade [41].

Apesar das vantagens, a segurança dos dados transmitidos é uma preocupação central, já que vulnerabilidades como ataques de interceptação podem comprometer a privacidade e a integridade das informações dos consumidores. Para proteger esses sistemas, medidas de cibersegurança robustas como criptografia de dados em trânsito e autenticação mútua entre dispositivos e servidores são essenciais para prevenir ameaças cibernéticas [42]. Além disso, a adoção em larga escala dos medidores inteligentes enfrenta desafios relacionados tanto à segurança quanto à privacidade. A coleta frequente e detalhada de dados de consumo residencial pode revelar padrões de comportamento, como horários de entrada e saída de casa, o que levanta preocupações éticas e legais sobre o uso dessas informações. Quando esses dados são transmitidos por meio de protocolos de comunicação com proteção insuficiente, ausência de criptografia, falhas na autenticação ou atualizações de segurança defasadas, o sistema se torna vulnerável a acessos não autorizados, vazamentos ou manipulação maliciosa [42]. Outro obstáculo é o custo de implementação para o consumidor final, especialmente quando o medidor não é fornecido pela concessionária. Mesmo nos casos em que a empresa de energia arca com o investimento inicial, o tempo de retorno econômico (payback) tende a ser longo, o que pode desestimular a adesão voluntária. Ademais, o funcionamento eficaz dos medidores inteligentes depende de uma infraestrutura robusta de comunicação incluindo conectividade estável e cobertura adequada. Essa exigência representa um desafio significativo, particularmente em regiões remotas ou com acesso limitado à internet. Diante disso, os investimentos em conectividade e redes de comunicação devem ser planejados de forma gradativa e estratégica, garantindo uma transição sustentável, equitativa e acessível para todos os usuários [43].

2.5 Tecnologias de Big Data e Machine Learning

O avanço das tecnologias de Big Data e Machine Learning tem transformado a maneira como organizações e indivíduos lidam com dados. Com o aumento exponencial de informações, é essencial desenvolver métodos eficientes de armazenamento, processamento e análise para extrair insights valiosos e apoiar a tomada de decisões [44]. Big Data refere-se ao processamento e armazenamento de grandes volumes de dados que apresentam características complexas e dinâmicas. Esse conceito é sustentado por três pilares fundamentais conhecidos como os "3 Vs": Volume, Velocidade e Variedade. Cada um desses aspectos desempenha um papel crucial no manejo eficiente dos dados como [45]:

- Volume: representa a escala massiva de informações geradas continuamente. No setor

Figura 4 – Os 3 Vs do Big Data

Fonte: Adptado de [34]

energético, por exemplo, dados provenientes de medidores inteligentes, sensores IoT e redes elétricas são produzidos em grande quantidade. Esses dados precisam ser armazenados e processados rapidamente para fornecer insights precisos que orientem operações e planejamentos.

- Velocidade: refere-se à taxa na qual os dados são gerados e a necessidade de processá-los em tempo real. No contexto da rede elétrica, essa característica é vital, uma vez que análises instantâneas permitem que operadoras monitorem o desempenho da infraestrutura e integrem fontes renováveis de energia, como solar e eólica, de forma ágil e eficiente.

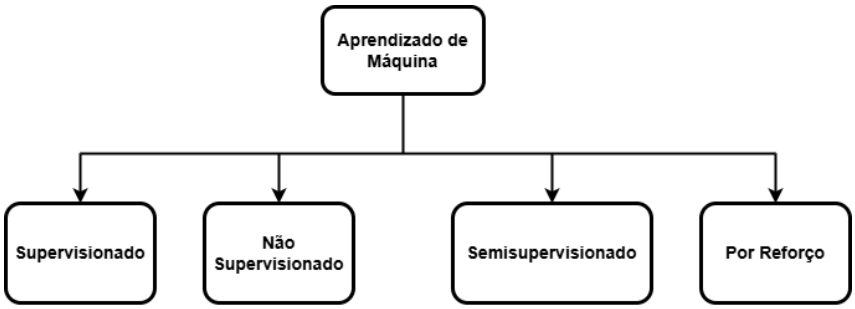
- Variedade: engloba a diversidade de formatos e tipos de dados disponíveis. Esses podem variar entre estruturados (como medições de consumo de energia), semi-estruturados (como logs de sensores) e não estruturados (como postagens em redes sociais). A capacidade de lidar com essa multiplicidade é essencial para realizar análises abrangentes e extrair valor máximo das informações.

No segmento energético, as tecnologias de Big Data têm se mostrado indispensáveis para concessionárias e operadores de rede. A coleta de dados de múltiplas fontes possibilita uma visão detalhada do consumo de energia e do comportamento da rede elétrica. Essas informações permitem identificar padrões de uso, prever tendências futuras e otimizar a alocação de recursos energéticos [46]. Além disso, a análise de dados em larga escala facilita a integração de fontes renováveis de energia, ajustando a oferta à demanda de forma dinâmica. Isso é particularmente relevante em cenários onde a produção de energia depende de condições climáticas variáveis, como vento e insolação. Ao utilizar Big

Data, operadores conseguem garantir maior eficiência energética e suportar o aumento do consumo e da complexidade das redes modernas [44].

Machine Learning, por sua vez, é uma subárea da Inteligência Artificial focada no desenvolvimento de modelos que aprendem e aprimoram suas funções com base em dados. Em vez de serem programados explicitamente para realizar tarefas específicas, esses modelos ajustam seus parâmetros e melhoram o desempenho à medida que novos dados são incorporados, permitindo previsões, identificação de padrões complexos e tomadas de decisão automatizadas [47]. O aprendizado de máquina é comumente classificado em quatro tipos principais: aprendizado supervisionado, não supervisionado, semi-supervisionado e aprendizado por reforço [48], conforme ilustrado na Figura 5.

Figura 5 – Classificação dos Algoritmos de Aprendizado de Máquina



Fonte: Autor(2025)

2.5.1 Aprendizado Supervisionado

No aprendizado supervisionado, os modelos são treinados utilizando dados rotulados, o que permite que o algoritmo aprenda a mapear entradas para as saídas desejadas com base em exemplos fornecidos. Esse processo envolve o uso de algoritmos como árvores de decisão, *classificadores Naive-Bayes*, máquinas de vetor de suporte, redes neurais e outros, cada um com suas particularidades para identificar padrões nos dados [48, 49]. Para entender completamente o aprendizado supervisionado, é importante distinguir dois tipos de variáveis: as variáveis independentes (ou preditoras), que representam as características de entrada, e a variável dependente (ou alvo), que é o valor a ser previsto ou estimado [48].

O objetivo do aprendizado supervisionado é desenvolver um modelo matemático capaz de associar as variáveis independentes à variável alvo. O modelo aprende os padrões presentes nos dados históricos e ajusta seus parâmetros de forma iterativa para aproximar a relação entre as entradas e as saídas. Após o treinamento, o modelo pode receber novos dados de entrada e gerar previsões precisas com base nos padrões aprendidos. Exemplos de aplicação incluem a previsão de preços de imóveis, a classificação de e-mails como spam ou não spam, e o diagnóstico de doenças a partir de exames médicos. Assim, o aprendizado

supervisionado desempenha um papel fundamental na criação de sistemas preditivos que têm impactos significativos em diversas áreas, como negócios e saúde [47].

2.5.2 Aprendizado Não Supervisionado

Aprendizagem não supervisionada é uma abordagem de Machine Learning em que o algoritmo é treinado em um conjunto de dados não rotulado, ou seja, dados para os quais as saídas desejadas não são fornecidas. Nesse contexto, o objetivo principal é explorar a estrutura e os padrões subjacentes nos dados sem orientação prévia sobre as saídas esperadas. Diferentemente do aprendizado supervisionado, onde o modelo recebe exemplos rotulados para aprender a fazer previsões, no aprendizado não supervisionado, o algoritmo tenta identificar padrões intrínsecos por conta própria, sem a necessidade de rótulos [47, 48].

As técnicas de aprendizado não supervisionado incluem métodos como o clustering com o algoritmo K-means, que agrupa os dados em clusters com base em similaridades, além de outras abordagens como modelos de mistura gaussiana, modelos ocultos de Markov e análise de componentes principais PCA (*Principal Component Analysis*). Essas técnicas são amplamente utilizadas em tarefas como segmentação de mercado, análise exploratória de dados e compressão de dados, permitindo uma exploração eficaz dos dados para a descoberta de padrões e relações complexas sem a necessidade de rótulos pré-existentes [49, 50].

2.5.3 Aprendizado Semi-Supervisionado

O aprendizado semi-supervisionado combina elementos do aprendizado supervisionado e não supervisionado para aproveitar tanto dados rotulados quanto dados não rotulados, uma vez que em muitas aplicações os dados rotulados são escassos ou de difícil obtenção. Nesse cenário, o método utiliza um pequeno conjunto de dados rotulados para guiar o treinamento, enquanto o restante dos dados não rotulados é explorado para descobrir estruturas ou padrões que ajudam a refinar o modelo. A vantagem desse método é a possibilidade de alcançar resultados mais precisos do que o aprendizado puramente não supervisionado, mas sem a necessidade de rotular grandes volumes de dados, reduzindo o custo e o esforço de anotação [47].

As técnicas utilizadas no aprendizado semi-supervisionado incluem modelos generativos, que tentam capturar a distribuição dos dados para auxiliar na predição das classes dos dados não rotulados, e máquinas de vetor de suporte transdutivo, que se concentram em classificar especificamente os dados disponíveis. Além disso, métodos baseados em grafos utilizam relações entre os dados para propagar rótulos dos dados rotulados para os não rotulados, enquanto métodos de auto-treinamento permitem que o modelo inicial,

treinado com dados rotulados, rotule iterativamente parte dos dados não rotulados para melhorar seu próprio desempenho. Essas abordagens são particularmente úteis em tarefas como reconhecimento de fala, processamento de linguagem natural e visão computacional, onde rotular grandes conjuntos de dados pode ser inviável [50, 51].

2.5.4 Aprendizado por Reforço

O aprendizado por reforço RL (*Reinforcement Learning*) é uma abordagem de aprendizado de máquina na qual um agente interage com um ambiente, toma decisões e recebe feedback na forma de recompensas ou penalidades. O objetivo principal é que o agente aprenda, ao longo do tempo, a realizar ações que maximizem as recompensas acumuladas, desenvolvendo uma política de ação eficiente para alcançar resultados ideais [52, 53]. Esse paradigma é particularmente eficaz em contextos que envolvem decisões sequenciais, como jogos, controle de robôs, navegação autônoma e sistemas de gerenciamento inteligente de energia. Por meio da exploração e da experimentação contínua, o agente ajusta suas estratégias com base nas consequências observadas de suas ações, promovendo um aprendizado adaptativo [54]. Entre as técnicas mais consagradas no aprendizado por reforço estão:

Q-learning: Um algoritmo de aprendizado baseado em valores que aprende a função de valor-ação para cada estado, utilizando uma política off-policy. Esse método foi formalmente introduzido por Watkins e Dayan [55].

DQN (*Deep Q-Networks*): Uma extensão do Q-learning que utiliza redes neurais profundas para representar a função de valor, tornando possível lidar com ambientes com estados de alta dimensionalidade. Essa técnica foi popularizada pelo trabalho de Mnih et al., que demonstraram controle em nível humano em jogos da Atari [56].

Por fim, técnicas de RL também têm sido amplamente aplicadas em sistemas inteligentes, como smart grids, para otimizar a gestão de energia, prever padrões de consumo e melhorar a tomada de decisões em tempo real [57].

2.6 Revisão de Trabalhos Relacionados

A revisão de trabalhos relacionados é essencial para identificar as contribuições existentes na área e posicionar a pesquisa no contexto do conhecimento atual. Diversos estudos têm investigado o uso do TinyML em diferentes contextos. A Tabela 4 resume e compara as principais características dos trabalhos relacionados discutidos nesta seção, destacando as contribuições e diferenciais da proposta apresentada neste trabalho em relação ao estado da arte.

Tabela 4 – Relação de trabalhos e suas contribuições

Referência	**TinyML ESP32**	**Energia Elétrica**	**Inferência Local**	**Inferência Nuvem**	**Consumo de Energia Otimizado**	**Quantificação Pós-Treinamento**
[58]	-	X	-	X	-	-
[59]	-	X	-	X	-	-
[60]	X	X	X	-	X	X
[61]	X	-	X	-	X	-
[62]	-	-	X	-	X	-
[63]	-	-	X	-	X	-
[64]	X	-	X	-	X	-
[65]	X	-	X	-	X	-
[66]	-	X	X	-	X	X
Este trabalho	X	X	X	-	X	X

Fonte: Autor (2025)

O estudo de [58] propôs um framework centralizado para detecção de fraudes em medidores inteligentes, mas com demandas altas de infraestrutura, custos elevados e latência. Em contraste, nosso trabalho utiliza TinyML no ESP32 para detectar anomalias localmente e em tempo real, eliminando a necessidade de conexão com a nuvem. Essa abordagem se mostra mais adequada para redes elétricas distribuídas e regiões com infraestrutura limitada. Em [59], Hayashi et al. (2020) exploraram uma arquitetura B2B (*Business-to-Business*)/B2C (*Business-to-Consumer*) para integração de IoT e ML em smart meters, mas sem aplicação embarcada. Nosso trabalho se diferencia por implementar e otimizar os modelos diretamente no dispositivo, possibilitando respostas imediatas e menor consumo de dados. Em [60], Aghera et al. (2021) propuseram uma abordagem para monitoramento energético não intrusivo que realiza a inferência localmente em dispositivos embarcados de baixo custo, como o ESP32. O método combina redes 1D-CNN (*One-Dimensional Convolutional Neural Network*) para identificar o estado dos aparelhos e LSTM (*Long Short-Term Memory*) para estimar seu consumo, operando sobre dados de baixa frequência (1 minuto) do dataset público REDD, que contém medições reais de consumo residencial. Essa estratégia reduz a dependência da internet e permite feedback em tempo real com baixo consumo de energia e alta autonomia do sistema.

O TinyML tem sido empregado em diversas áreas, como o monitoramento agrícola apresentado por [61], onde drones com TinyML classificaram doenças em plantações de tomate com 92,30% de precisão, destacando a eficiência da inferência local. Inspirados por abordagem como essa, aplicamos o TinyML no setor energético para monitoramento de consumo e detecção de anomalias em smart meters, garantindo processamento rápido e baixa latência sem depender da nuvem.

No contexto de eficiência energética, [62] implementaram uma solução de processamento de dados na borda para monitoramento da saúde de animais. Utilizando uma

rede LoRa e um microcontrolador otimizado, o estudo conseguiu aumentar a autonomia do dispositivo de 13 para 39 dias. Esse resultado demonstra como o aprendizado de máquina embarcado pode ser utilizado para reduzir significativamente o consumo energético, garantindo maior autonomia dos dispositivos IoT.

A detecção de anomalias estruturais também tem sido abordada com soluções inovadoras. No trabalho de [63], foi desenvolvido um modelo de detecção de anomalias em microcontroladores de baixa potência (STM32L4) usando Análise de Componentes Principais e Autoencoders. A solução reduziu o tráfego de rede, transmitindo menos de 10 Bytes/hora por instalação, e foi validada em uma ponte na Itália, demonstrando eficácia na redução de custos operacionais e no suporte à manutenção preditiva de infraestruturas críticas.

No campo do monitoramento ambiental [64], Wardana et al. (2023) desenvolveram um dispositivo de baixo custo para avaliação da qualidade do ar, utilizando TinyML no Raspberry Pi Pico W. O sistema, alimentado por energia solar, obteve um coeficiente de determinação (R^2) acima de 0,70 na previsão de variáveis ambientais e apresentou 80% de eficácia na imputação de dados ausentes, demonstrando a viabilidade de soluções TinyML para monitoramento sustentável. Enquanto que em [65], Hayajneh et al. (2024) propuseram um sistema inteligente que combina TinyML e Transfer Learning para previsão da umidade do solo e otimização da irrigação. Sensores IoT com ESP32 processam localmente os dados por meio de modelos LSTM e DNN (*Deep Neural Network*), reduzindo a dependência da internet. Além disso, drones são utilizados para atualizar os modelos via Wi-Fi, aumentando a autonomia e eficiência do sistema. A proposta contribui para a redução do consumo de energia e desperdício de água, além de promover o aumento da produtividade agrícola. Complementarmente em [66] , Rashid et al. (2019) aplicaram técnicas de quantização para reduzir o custo computacional de modelos preditivos, viabilizando sua execução eficiente em dispositivos embarcados com recursos limitados.

Um dos principais gaps identificados na literatura é a ausência de soluções integradas de monitoramento energético com inferência embarcada que atendam adequadamente às exigências de segurança e eficiência em medidores inteligentes, evidenciado por meio do mapeamento sistemático em [67]. Diante disso, este trabalho **propõe uma solução escalável e de baixo custo, baseada em modelos de machine learning otimizados e validados em cenário real. O diferencial reside na comparação sistemática entre três abordagens distintas XGBoost, One-Class SVM e Autoencoder avaliadas quanto à precisão, uso de memória e tempo de inferência, todas testadas diretamente no microcontrolador ESP32**. Ao integrar IoT, TinyML e algoritmos otimizados, a solução apresentada avança significativamente em relação ao estado da arte, oferecendo um sistema prático, seguro e eficiente para monitoramento e análise preditiva do consumo energético em tempo real.

3 Fundamentação Teórica

Esta capítulo apresenta os conceitos fundamentais que embasam o desenvolvimento e a aplicação de modelos de aprendizado de máquina em dispositivos embarcados, com foco na área emergente do TinyML. São discutidos os principais conceitos, arquiteturas e aplicações dessa tecnologia, bem como as técnicas de otimização utilizadas para adaptar modelos de Machine Learning a ambientes com recursos computacionais limitados. Por fim, são abordadas as principais limitações e desafios enfrentados na implementação prática do TinyML, especialmente em dispositivos de Internet das Coisas (IoT) e sistemas embarcados. Essa base teórica é essencial para compreender o contexto e a viabilidade do uso do TinyML em cenários reais.

3.1 Análise Preditiva: Conceitos e Técnicas

A análise preditiva é um ramo da ciência de dados que utiliza técnicas de estatística e machine learning para analisar dados históricos e fazer previsões sobre eventos futuros. O objetivo da análise preditiva é identificar padrões nos dados que possam ser utilizados para antecipar comportamentos, tendências e resultados futuros. Essa abordagem é amplamente aplicada em diversas áreas, desde finanças e saúde até marketing e logística, permitindo decisões mais informadas e estratégicas [68].

As técnicas de análise preditiva variam desde métodos simples, como regressões lineares e médias móveis, até algoritmos complexos DL (*Deep Learning*). Uma das técnicas mais comuns é a modelagem de séries temporais, que analisa dados ao longo do tempo para identificar padrões sazonais e tendências de longo prazo. Além disso, técnicas de aprendizado supervisionado e não supervisionado são amplamente utilizadas para treinar modelos capazes de identificar padrões nos dados. A escolha da técnica adequada depende da natureza dos dados e do problema a ser solucionado, assim como dos recursos computacionais disponíveis para a análise [69].

Um dos principais benefícios da análise preditiva é a sua capacidade de fornecer dados em tempo real. Isso é possível graças ao uso de algoritmos que processam dados de forma contínua, permitindo ajustes rápidos e dinâmicos conforme novas informações surgem. Além disso, a análise preditiva pode ser combinada com simulações de cenários, permitindo que gestores avaliem diferentes estratégias antes de implementá-las. Essa abordagem é especialmente valiosa em ambientes de alta variabilidade, onde as mudanças precisam ser gerenciadas de forma eficiente para garantir resultados consistentes e sustentáveis [70].

3.2 Modelos de Machine Learning para Análise e Previsão de Consumo Energético.

Modelos de *machine learning* desempenham um papel fundamental na previsão de consumo de energia, permitindo a análise de grandes volumes de dados e a identificação de padrões complexos que seriam difíceis de capturar com métodos tradicionais. Esses modelos utilizam dados históricos para aprender as relações entre variáveis, gerando previsões mais precisas, além de auxiliar na detecção de anomalias e na otimização de sistemas.

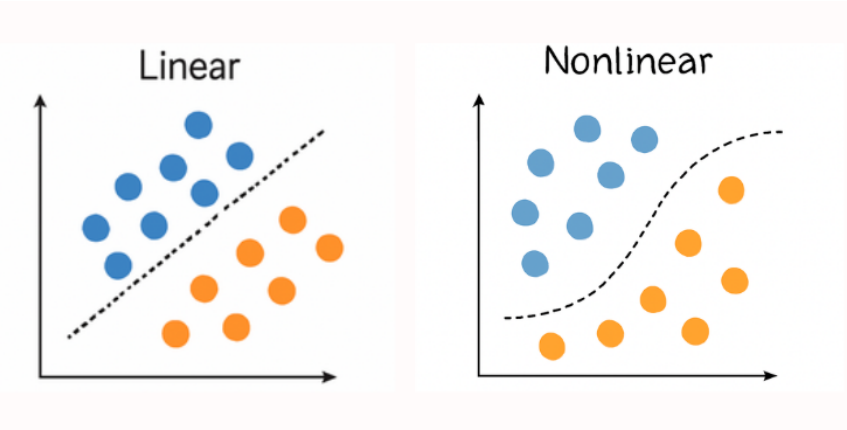
Neste trabalho, o objetivo é implementar um sistema de monitoramento e previsão de consumo energético, utilizando machine learning para identificar padrões de consumo, prever demandas futuras, detectar anomalias e apoiar a manutenção preditiva em redes inteligentes. Em cenários de redes de energia, onde os dados de consumo apresentam variabilidade sazonal, flutuações complexas e possíveis eventos anômalos, a escolha adequada dos algoritmos é essencial para garantir a precisão, eficiência computacional e adaptabilidade do sistema. Para atingir esses objetivos, foram selecionados três algoritmos principais: *One-Class SVM*, *XGBoost Regressor* e *Autoencoder*. Cada um foi escolhido com base em suas capacidades específicas para lidar com os desafios característicos do domínio energético. A seguir, são apresentados os algoritmos que compõem o protótipo desenvolvido.

3.3 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte proposta por Vapnik et al. em 1992 [71], é um algoritmo de aprendizado de máquina supervisionado amplamente utilizado para tarefas de classificação e regressão. Seu principal objetivo é encontrar um hiperplano ótimo que separa diferentes classes no espaço de características, maximizando a margem entre elas. Este modelo pode ser estendido para resolver problemas não linearmente separáveis por meio do uso de uma técnica chamada kernel trick, que mapeia os dados para um espaço de maior dimensão, onde as classes se tornam linearmente separáveis. Como ilustrado na figura 6, tem-se dois exemplos práticos de SVM: à esquerda, é mostrado um problema linearmente separável, e à direita, um exemplo não linearmente separável, com o uso do kernel.

A ideia central da SVM é encontrar a linha ou hiperplano de separação que maximize a margem de separação entre as classes. A margem é a distância entre o hiperplano de separação e os vetores de suporte, que são os pontos de dados mais próximos das classes. O modelo visa maximizar essa margem para aumentar a generalização do modelo, tornando-o mais robusto a ruídos nos dados e evitando o overfitting [73].

Figura 6 – Classes Linearmente e Não Linearmente Separáveis



Fonte: [72]

3.3.1 One Class SVM

One-Class Support Vector Machine (One-Class SVM) é uma variante da SVM tradicional, projetada para a detecção de anomalias ou *outliers*, quando apenas amostras da classe normal estão disponíveis para treinamento [74]. Diferentemente da SVM convencional, que realiza classificação supervisionada entre múltiplas classes, a One-Class SVM é um método não supervisionado que busca delimitar a região que contém a distribuição dos dados da classe-alvo, possibilitando a identificação de pontos que se desviam desse padrão.

Anomalias são observações que se desviam significativamente do comportamento esperado, podendo indicar falhas, fraudes ou eventos raros e inesperados. Exemplos típicos incluem detecção de fraudes financeiras, monitoramento de falhas em sistemas e identificação de padrões inéditos [75]. Em cenários onde apenas dados normais estão disponíveis, a One-Class SVM destaca-se por sua capacidade de modelar essa classe e reconhecer dados atípicos.

A principal diferença entre a SVM tradicional e a *One-Class SVM* está nos objetivos e aplicações de cada modelo. Enquanto a SVM clássica busca encontrar um hiperplano que separe múltiplas classes com margem máxima, a One-Class SVM tem como foco modelar a distribuição de uma única classe, identificando instâncias que se afastam do padrão esperado. Essa característica a torna especialmente útil tanto para detecção de anomalias quanto para identificação de novidades [72].

A *One-Class SVM* atua ao maximizar a margem que separa os dados da classe normal do restante do espaço de características, sendo particularmente eficaz em cenários com dados desbalanceados, onde as anomalias são raras e de difícil rotulação. Sua aplicação tem se mostrado bem-sucedida também em domínios com alta dimensionalidade, como em sistemas de IoT e monitoramento contínuo de sensores, destacando-se

pela robustez e precisão frente a outros métodos de detecção não supervisionados [76].

Em suma, a *One-Class SVM* representa uma abordagem poderosa para a detecção de anomalias em grandes volumes de dados, sendo aplicada com eficiência em ambientes complexos e heterogêneos, onde a identificação de comportamentos anômalos é crítica para a segurança e a manutenção dos sistemas.

3.3.2 XGBoost Regressor

O *XGBoost Regressor* é amplamente reconhecido como um dos algoritmos mais eficazes no campo do aprendizado de máquina, especialmente para tarefas de regressão e classificação. Ele pertence à categoria de métodos de aprendizado em conjunto baseados em árvores, como o GBM (*Gradient Boosting Machines*) [77]

Desenvolvido por Tianqi Chen e Carlos Guestrin na Universidade de Washington, o XGBoost foi lançado oficialmente em 2016 com a proposta de ser uma implementação otimizada e de alto desempenho do algoritmo Gradient Boosting. O foco do XGBoost está em três pilares principais: Velocidade, Precisão e Escalabilidade. O algoritmo opera combinando várias árvores de decisão para formar um modelo mais robusto, sendo capaz de gerar árvores profundas de forma iterativa, o que resulta em modelos altamente sofisticados [78]. O princípio do XGBoost é baseado em *Gradient Boosting*, uma técnica que visa aprimorar a precisão das previsões, adicionando técnicas de regularização para evitar o *overfitting*, além de otimizar o tempo de treinamento. Dentre suas principais inovações, destacam-se a introdução de uma técnica para a determinação do melhor valor para um nó, utilizando a pontuação de semelhança e o ganho. Também se destaca o uso de regularização com os parâmetros λ (semelhante ao decaimento de peso em redes neurais) e γ (poda de árvores).

Uma das maiores vantagens do XGBoost é sua capacidade de fazer boosting paralelo, o que permite que ele seja aplicado de forma eficiente em grandes volumes de dados. Ele tem mostrado grande sucesso em diversos cenários práticos, como no caso de [79] e [80], onde foi utilizado para detectar fraudes em consumidores de energia elétrica e água. Embora o modelo ofereça um desempenho superior ao de uma única árvore de decisão, ele sacrifica parte da interpretabilidade, característica natural das árvores de decisão simples. Para mitigar essa limitação, podem ser aplicadas técnicas de compressão de modelos, permitindo representar o XGBoost por uma única árvore que aproxime as decisões do modelo original. O funcionamento do XGBoost segue uma estrutura baseada em árvores de decisão otimizadas por gradiente, envolvendo as seguintes etapas [78]:

- **Seleção do Atributo**: Inicialmente, o algoritmo escolhe um atributo relevante para a tarefa de classificação ou regressão a ser realizada.

- **Divisão dos Dados**: O conjunto de dados é dividido em subconjuntos, com base no atributo selecionado, de forma que cada grupo contenha amostras com valores específicos para o atributo escolhido.
- **Critério de Divisão**: O algoritmo usa um critério para dividir os dados, como ganho de informação, índice Gini ou erro de classificação, visando encontrar a divisão que melhor separa as classes.
- **Recursão**: Esse processo de divisão é repetido recursivamente nos nós filhos, criando uma estrutura hierárquica de árvore. O processo continua até que seja atingido um critério de parada, como uma profundidade máxima ou um número mínimo de amostras em um nó.
- **Classificação ou Regressão**: Quando a árvore é construída, ela é usada para fazer previsões em novos dados. Esses dados seguem o caminho na árvore de acordo com os valores de seus atributos, até chegar à folha correspondente à classe ou valor predito.

3.3.3 AutoEncoders

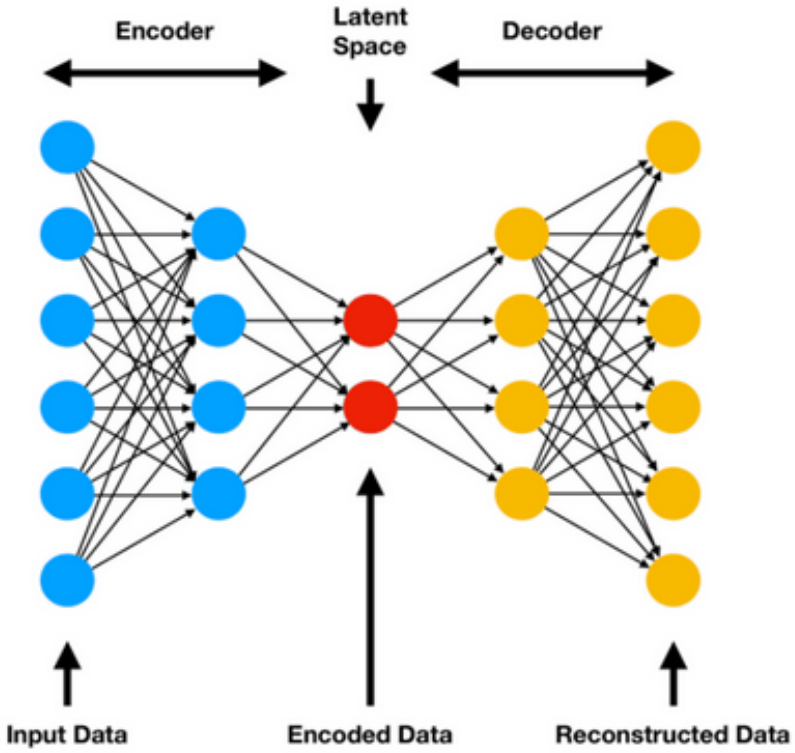
Os AE (*Autoencoder*) foram inicialmente propostos como uma solução não linear para análise de componentes principais, conforme descrito por Kramer (1991) [81]. Desde então, sua aplicação expandiu-se para diversas áreas, incluindo redução de dimensionalidade, remoção de ruído, detecção de anomalias e até mesmo geração de novos dados semelhantes aos de treinamento. Sua principal característica é a capacidade de aprender representações compactas e eficientes dos dados de entrada. Um *Autoencoder* é um tipo de rede neural usada em aprendizado não supervisionado, cujo objetivo principal é aprender uma representação comprimida dos dados de entrada [82, 83]. Ele é composto por duas partes principais:

- **Codificador:** O codificador recebe os dados de entrada e os comprime em uma representação no espaço latente, criando uma versão reduzida e compactada dos dados. Essa representação, de menor dimensão, facilita a extração de características relevantes e a compressão eficiente da informação.
- **Decodificador:** O decodificador tem a função de reconstruir os dados originais a partir dessa representação comprimida. O objetivo é gerar uma saída tão próxima quanto possível da entrada original, minimizando a diferença entre elas.

O treinamento de um Autoencoder envolve a minimização do erro de reconstrução — isto é, a diferença entre os dados de entrada e a saída reconstruída pelo modelo. Esse

processo é realizado iterativamente até que o modelo aprenda uma representação eficaz dos dados. A figura 7 ilustra a arquitetura básica de um Autoencoder.

Figura 7 – Arquitetura de um AutoEncoder



Fonte: [84]

3.3.4 Autoencoders para Detecção de Anomalias

Os Autoencoders têm se mostrado altamente eficazes na detecção de anomalias, especialmente em sistemas com dados complexos e alta dimensionalidade, como séries temporais. Esses modelos são treinados exclusivamente com dados normais, o que lhes permite aprender a representação característica do comportamento considerado "normal". Durante a fase de inferência, o modelo tenta reconstruir novas entradas com base nesse padrão aprendido, e falhas significativas na reconstrução indicam possíveis anomalias [85].

No treinamento, o autoencoder recebe apenas exemplos de dados normais, buscando minimizar a discrepância entre as entradas originais e suas reconstruções. Para isso, costuma-se utilizar uma função de perda como o Erro Quadrático Médio ou MSE (*Mean Squared Error*).

Na fase de inferência, quando o modelo recebe novos dados, ele tentará reconstruí-los com base no padrão aprendido. Se os dados forem normais, a reconstrução tende a ser precisa, resultando em um erro baixo. Já se forem anômalos ou seja, substancialmente diferentes do padrão aprendido, o erro de reconstrução será significativamente maior.

Para realizar a detecção de anomalias, define-se um limiar para o erro de reconstrução. Valores acima desse limiar são considerados indicativos de comportamento anômalo. Esse procedimento torna os autoencoders particularmente úteis para identificar padrões atípicos em diversos domínios, como séries temporais de consumo energético ou dados de telemetria de satélites [86].

Segundo Sakurada e Yairi (2014), os autoencoders são capazes de detectar anomalias sutis que passam despercebidas por métodos lineares, como a Análise de Componentes Principais. Além disso, os autores destacam que esses modelos conseguem aprender com eficiência a estrutura subjacente aos dados normais, respondendo de forma distinta ao processarem entradas anômalas [87].

Assim, a arquitetura dos autoencoders consiste em duas partes principais: um codificador, responsável por mapear os dados de entrada para um espaço latente de menor dimensão, e um decodificador, que tenta reconstruir os dados originais a partir dessa representação comprimida. O treinamento é realizado de forma não supervisionada, permitindo que o modelo aprenda padrões relevantes sem a necessidade de rótulos.

3.4 TinyML: Conceitos, Arquiteturas e Aplicações

O TinyML é um campo do aprendizado de máquina que possibilita a execução eficiente de modelos em dispositivos com recursos computacionais limitados, como microcontroladores e sensores embarcados. Essa capacidade permite realizar inferência localmente, dispensando a necessidade de enviar dados para servidores centrais, o que reduz a latência, aumenta a privacidade e viabiliza aplicações em ambientes com conectividade limitada ou intermitente [88].

As arquiteturas de TinyML são projetadas para otimizar o uso de memória e energia nos dispositivos de implementação. Plataformas como TensorFlow Lite for Microcontrollers e Edge Impulse desempenham um papel fundamental na popularização dessa tecnologia, pois simplificam a conversão de modelos tradicionais de machine learning em versões compactas e leves, adequadas para integração em dispositivos IoT [89]. Essas ferramentas permitem executar modelos em plataformas como ESP32 e Arduino, possibilitando análises em tempo real no campo, aplicáveis ao monitoramento ambiental, detecção de falhas em equipamentos e análise de consumo de energia [90].

No contexto da IoT, o TinyML é especialmente relevante por permitir o processamento local de grandes volumes de dados em tempo real. Suas aplicações incluem detecção de sons, reconhecimento de imagens, monitoramento ambiental e análise de padrões de consumo em medidores inteligentes. Sua principal vantagem reside na capacidade de operar em ambientes com conectividade limitada ou inexistente, assegurando autonomia e funcionalidade mesmo sem acesso à nuvem [88].

A eficiência energética também é um diferencial importante, tornando o *TinyML* ideal para dispositivos alimentados por baterias, prolongando sua vida útil e garantindo operações contínuas [91]. Em cenários de energia renovável, como fazendas solares e parques eólicos, ele pode ser usado para monitorar o estado dos equipamentos e prever necessidades de manutenção, contribuindo para maior eficiência e confiabilidade operacional [92]. A aplicação de *TinyML* oferece diferenciais significativos ao possibilitar a execução eficiente de modelos de aprendizado de máquina em dispositivos com recursos mínimos. Entre os principais benefícios, destacam-se:

- **Processamento em Tempo Real:** *TinyML* permite o processamento de dados em tempo real diretamente nos dispositivos, sem a necessidade de transmitir grandes volumes de dados para servidores centrais. Isso reduz a latência, permitindo respostas imediatas a eventos e mudanças no consumo de energia [93].
- **Redução de Latência e Dependência de Conectividade:** Como os algoritmos de *machine learning* são executados localmente nos dispositivos IoT, a dependência da conectividade com a nuvem é minimizada. Isso é crucial para cenários onde a conectividade é limitada ou intermitente, garantindo que o sistema continue a funcionar de forma eficiente [94].
- **Eficiência Energética e Baixo Consumo de Recursos:** Dispositivos *TinyML* são projetados para operar com baixo consumo de energia, tornando-os ideais para aplicações em *smart meters* e outros dispositivos IoT onde a eficiência energética é uma preocupação [93].
- **Detecção de Anomalias e Fraudes:** A capacidade de analisar dados em tempo real permite a detecção imediata de anomalias e possíveis fraudes no consumo de energia. Algoritmos de *machine learning* embarcados podem identificar padrões incomuns de consumo e alertar os operadores para ações corretivas.
- **Escalabilidade e Flexibilidade:** A implementação de *TinyML* permite a escalabilidade do sistema de monitoramento, uma vez que novos dispositivos podem ser facilmente integrados e configurados para executar modelos de *machine learning*. Isso proporciona flexibilidade na expansão e atualização do sistema sem grandes investimentos em infraestrutura.
- **Privacidade e Segurança dos Dados:** Processar dados localmente nos dispositivos IoT reduz os riscos associados à transmissão de dados sensíveis pela rede. Isso melhora a privacidade e segurança das informações dos consumidores, atendendo a regulamentações e expectativas de proteção de dados.

Em termos de aplicações, o TinyML é amplamente utilizado no processamento de dados em tempo real em sensores e atuadores inteligentes [95]. Esse processamento local

também é essencial para sistemas de monitoramento contínuo de saúde e reconhecimento de atividades em dispositivos vestíveis (wearables), onde a privacidade é aprimorada ao evitar comunicações constantes com a nuvem [96, 97]. Além disso, aplicações de visão computacional também se beneficiam do TinyML. Modelos de deep learning compactos, adaptados para rodar em dispositivos de baixo consumo de energia, são utilizados em classificação de imagens, detecção de objetos e reconhecimento de gestos. Exemplos práticos dessas implementações são encontrados em Paul et al. [98] e Mohan et al. [99], que demonstram como redes neurais convolucionais ou CNN (*Convolutional Neural Network*) podem ser otimizadas para funcionar em dispositivos de borda com memória extremamente limitada.

Essas capacidades tornam o TinyML ideal para implantação em dispositivos IoT, onde a eficiência energética, o processamento local de dados e a minimização da dependência de conectividade são elementos fundamentais para garantir operação contínua e eficiente dos sistemas.

3.4.1 Otimização de Modelos de Machine Learning

A otimização de modelos de Machine Learning é um aspecto crucial no contexto de TinyML, onde os modelos precisam ser alocados em dispositivos com recursos limitados de hardware. Nesse cenário, diversas técnicas de otimização são empregadas para garantir que os modelos sejam eficientes sem comprometer a precisão. A seguir, são apresentadas algumas das técnicas mais relevantes para essa otimização.

3.4.1.1 Quantização

A quantização é uma técnica fundamental em *TinyML*, que reduz a precisão dos parâmetros de um modelo (por exemplo, de 32 bits para 8 bits). Essa redução diminui a complexidade computacional sem causar uma perda significativa de desempenho [100], como ilustrado na figura 8. Existem dois métodos principais de quantização:

- **Quantização Pós-Treino:** Esse método envolve a representação dos modelos com menor precisão após o treinamento. Aqui, os pesos e as ativações do modelo são convertidos de floats de 32 bits para inteiros de 8 bits, otimizando o uso de memória e recursos computacionais sem a necessidade de retrainamento [101].
- **Treinamento Consciente de Quantização:** Neste método, a quantização é aplicada durante o treinamento do modelo. A diferença aqui é que, enquanto o processo *forward* do modelo (passagem da entrada à saída) é alterado para incorporar a quantização, o processo *backward* (ajuste dos pesos) permanece inalterado, preservando a integridade do treinamento. Isso permite uma adaptação mais eficiente do modelo às limitações de hardware [102].

Figura 8 – Exemplo de quantização de uma matriz de pesos de float32 para uint8

Fonte: [102]

3.4.1.2 Pruning

Pruning, que significa poda em português, é uma técnica fundamental para melhorar a eficiência dos modelos em TinyML. Essa técnica consiste na remoção de conexões e neurônios menos relevantes em redes neurais, reduzindo o número de parâmetros e simplificando o modelo, o que resulta em menor consumo de memória e menor custo computacional [103].

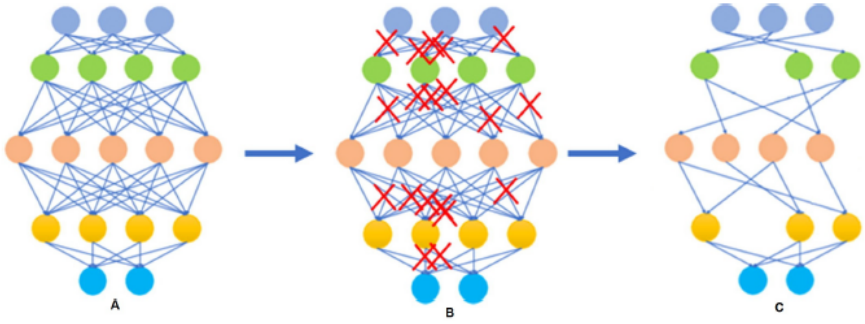
Existem diferentes abordagens para a poda, entre as quais destaca-se o *Pruning Simples*. Neste método, eliminam-se os pesos cujo valor é muito baixo (tipicamente próximos de zero), partindo da premissa de que tais pesos têm impacto mínimo no desempenho do modelo. A poda é realizada de forma gradual, iniciando-se com uma esparsidade inicial e avançando até uma esparsidade final desejada, que pode ser ajustada conforme a necessidade do sistema [104]. O processo de aumento da esparsidade ao longo do tempo é controlado por uma função de decaimento polinomial, expressa pela seguinte equação:

$$S = S_0 + (S_f - S_0) \cdot \left(\frac{t - t_0}{t_f - t_0} \right)^{\alpha} \tag{3.1}$$

onde:

- S é a esparsidade do modelo;
- S_f é a esparsidade final;
- S_0 é a esparsidade inicial;
- t é a etapa de tempo atual;
- t_f é a etapa final;
- t_0 é a etapa inicial;
- α é o expoente (geralmente 3 [105]).

Figura 9 – Exemplo ilustrativo do processo de pruning



Fonte: Adptado de [106]

Na figura 9 é exemplificado o conceito de *pruning*. No painel A, observa-se uma rede neural totalmente conectada. No painel B, ocorre a remoção gradual das sinapses (conexões) e neurônios menos importantes para o desempenho do modelo, processo este que visa eliminar redundâncias. Por fim, o painel C mostra a nova rede neural, reduzida em tamanho, com menos conexões e neurônios, mantendo a capacidade de aprendizado e inferência, porém com maior eficiência computacional e menor consumo de recursos.

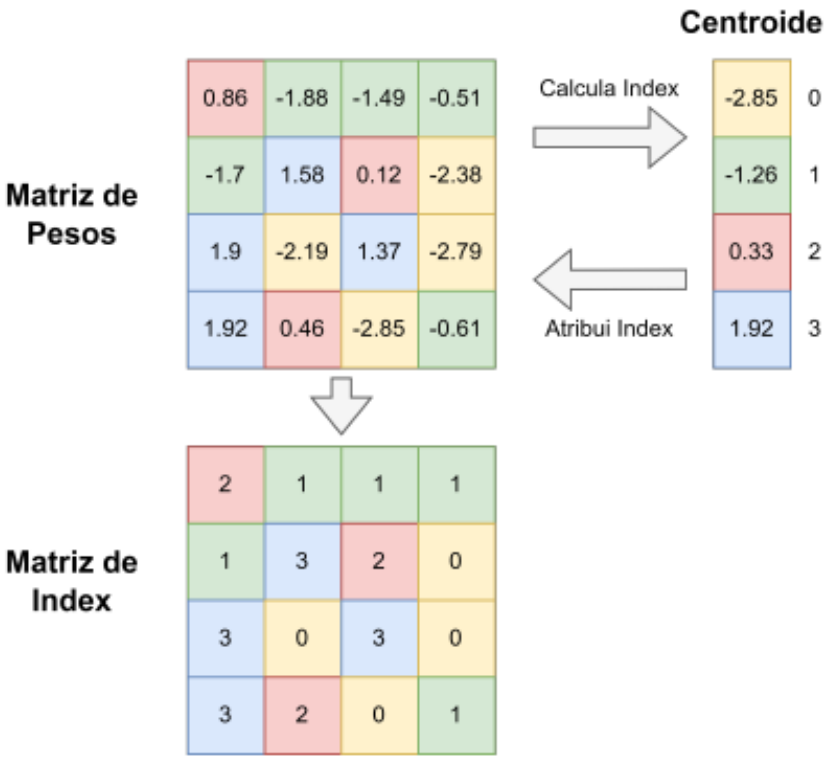
3.4.1.3 Agrupamento de Pesos

O agrupamento de pesos, também conhecido como weight clustering, é uma técnica que reduz o número de valores de peso únicos em uma rede neural, trazendo benefícios significativos para a implementação em dispositivos com recursos limitados, como microcontroladores usados em TinyML. O processo consiste em agrupar os pesos de cada camada do modelo em N clusters, atribuindo a cada peso o valor do centróide (média) do seu respectivo grupo. Dessa forma, vários pesos diferentes são representados por um único valor, o que diminui a variabilidade e o tamanho total do modelo, reduzindo o uso de memória e o custo computacional [107].

Na Figura 10 é ilustrado o processo de agrupamento de pesos. Na imagem, uma matriz de pesos original, contendo valores contínuos e variados, é representada por uma matriz colorida à esquerda. Cada cor indica pesos que serão agrupados. O processo segue duas etapas principais:

- **Cálculo dos centroides:** os pesos são agrupados em N clusters, e para cada cluster é calculado um valor representativo, o centroide, que é a média dos pesos daquele grupo. Os centroides são apresentados em uma matriz coluna à direita.
- **Atribuição dos índices:** cada peso da matriz original é substituído pelo índice do centroide ao qual pertence, formando a matriz de índices mostrada na parte inferior da figura 10.

Figura 10 – Agrupamento de pesos de uma camada de uma rede neural artificial.



Fonte: Adaptado de [107]

Esse método possibilita que, durante a inferência, os pesos não sejam armazenados individualmente, mas sim como índices que apontam para valores únicos, otimizando a representação do modelo. Essa otimização é crucial para dispositivos com baixa capacidade de memória e processamento, permitindo que algoritmos de aprendizado de máquina sejam executados de maneira eficiente em ambientes restritos.

3.5 Limitações e Desafios na Implementação do TinyML

A implementação do *TinyML* em dispositivos IoT e de ponta oferece vantagens inegáveis, porém não se configura como uma solução ideal para todas as situações. A implantação nessas plataformas enfrenta desafios complexos que impactam diretamente o desempenho e a viabilidade dos modelos de aprendizado de máquina em ambientes com recursos limitados. Uma das principais barreiras é o consumo de energia, uma vez que dispositivos de ponta operam com restrições energéticas significativas, dificultando a execução eficiente de modelos complexos de *ML* [108]. Além disso, a necessidade de modelos compactos é fundamental para atender às limitações de armazenamento e memória desses dispositivos. Embora técnicas como quantização e compressão de modelos ajudem a reduzir o tamanho, elas podem acarretar compromissos na precisão dos resultados. Dessa forma, encontrar o equilíbrio ideal entre o tamanho do modelo e seu desempenho perma-

nece um desafio constante para os profissionais que trabalham com *TinyML*, tornando a otimização eficaz dos modelos um aspecto essencial do processo [88].

Outro desafio relevante é a ausência de padronização em *frameworks* e *benchmarks* específicos para *TinyML*, o que gera inconsistências e problemas de compatibilidade entre diferentes dispositivos e aplicações. A criação de padrões para ferramentas, protocolos e métricas de avaliação é essencial para facilitar a implantação e garantir a interoperabilidade entre plataformas diversas [109].

A latência é uma preocupação crítica em aplicações que exigem respostas em tempo real. Devido às limitações de processamento e memória dos dispositivos de ponta, alcançar baixa latência representa um desafio significativo. Para mitigar essa limitação, técnicas de otimização algorítmica são frequentemente combinadas com arquiteturas híbridas de computação em nuvem e na borda, nas quais tarefas menos complexas são executadas localmente, enquanto operações mais intensivas são transferidas para servidores na nuvem [110].

Superar esses desafios requer o uso de técnicas inovadoras de otimização, o desenvolvimento de hardware eficiente e a criação de padrões universais que viabilizem a ampla adoção do *TinyML* em diferentes cenários da IoT e da computação de borda. A crescente demanda por algoritmos e arquiteturas otimizadas, bem como por ferramentas e *frameworks* padronizados que acelerem o processo de desenvolvimento e implantação, é essencial para o avanço do campo. Além disso, a integração do *TinyML* em sistemas já existentes exige conhecimento especializado tanto em aprendizado de máquina quanto em sistemas embarcados, o que ressalta a importância de recursos educacionais adequados e programas de capacitação específicos [111].

3.6 Conceitos de Energia Elétrica

Para compreender e interpretar corretamente os dados provenientes de sistemas de medição inteligente, é essencial dominar os principais conceitos relacionados à energia elétrica. No contexto de smart meters, termos como potência ativa, reativa e aparente, bem como corrente, tensão e fator de potência, desempenham um papel fundamental na caracterização do consumo energético. Esses parâmetros não apenas permitem o monitoramento eficiente dos perfis de consumo, mas também viabilizam a identificação de irregularidades operacionais e a previsão de demandas futuras. A seguir, cada conceito é detalhado com as respectivas expressões matemáticas.

3.6.1 Parâmetros Elétricos

O cálculo da potência elétrica é essencial tanto na análise de circuitos simples quanto em aplicações mais complexas, como na eletrônica de potência. Através dessa

grandeza, torna-se possível avaliar a eficiência de cargas e dispositivos conversores como inversores e retificadores além de quantificar o consumo de energia. No contexto deste trabalho, os parâmetros elétricos permitem converter o consumo energético em valores monetários, indicando o custo associado ao uso da eletricidade [112]. Em sistemas com corrente alternada, a potência elétrica é normalmente classificada em quatro categorias principais: potência instantânea, potência aparente, potência ativa e potência reativa (também chamada de não ativa).

3.6.1.1 Potência Instantânea

A potência instantânea é calculada pelo produto entre os valores da tensão e da corrente em cada instante de tempo, conforme representado na Equação 3.2 [112]:

$$p(t) = v(t) \cdot i(t), \tag{3.2}$$

onde:

- $p(t)$ = é a potência instantânea em watts no instante t;
- $v(t)$ = é o valor da tensão em volts no instante t;
- $i(t)$ = é a corrente em amperes no instante t.

É importante destacar que a Equação 3.2 é válida para sinais em tempo contínuo, nos quais a variável temporal t pode assumir infinitos valores. Em sistemas digitais ou amostrados, utiliza-se a representação em tempo discreto, na qual o tempo é substituído pela variável n, representando o índice das amostras. Assim, a potência instantânea é expressa por:

$$p[n] = v[n] \cdot i[n], \tag{3.3}$$

onde:

- $p[n]$ = é a potência instantânea em watts na amostra n;
- $v[n]$ = é o valor da tensão em volts na amostra n;
- $i[n]$ = é a corrente em amperes na amostra n.

3.6.1.2 Potência Reativa

Potência Reativa (Q): é a potência que oscila entre a fonte e a carga, armazenada temporariamente em campos magnéticos e elétricos, presentes em componentes como indutores e capacitores [112]. Ela é medida em volt-amperes reativos (VAR) e calculada por:

$$Q = V \times I \times \sin(\theta), \tag{3.4}$$

onde:

- Q = potência reativa (VAR);
- V = tensão (V);
- I = corrente elétrica (A);
- $\sin(\theta)$ = seno do ângulo de fase entre corrente e tensão.

As cargas indutivas causam um atraso na corrente em relação à tensão, enquanto as cargas capacitivas provocam um adiantamento da corrente em relação à tensão. Esse desfasamento afeta diretamente o fator de potência da instalação. Um valor baixo de fator de potência, geralmente abaixo de 0,9, indica um elevado consumo de potência reativa, o que pode resultar em sobrecarga dos condutores, aumento das perdas elétricas e acionamento inadequado dos dispositivos de proteção. Portanto, o gerenciamento adequado da potência reativa, por meio da correção do fator de potência, é essencial para garantir maior eficiência e segurança no sistema elétrico [113].

3.6.1.3 Potência Aparente

Potência Aparente (S): Representa a potência total fornecida à carga, sendo a soma vetorial das potências ativa e reativa. É medida em volt-amperes (VA) e pode ser calculada por:

$$S = V \times I \quad \text{ou} \quad S = \sqrt{P^2 + Q^2}, \tag{3.5}$$

onde:

- S = Potência aparente (VA)
- P = Potência ativa (W)

Esses conceitos são fundamentais para avaliar a demanda de energia e entender a utilização e distribuição de energia na rede elétrica.

3.6.1.4 Fator de Potência

O fator de potência indica a eficiência com que a energia elétrica é utilizada. Valores próximos a 1 representam um uso eficiente, enquanto valores baixos indicam uma presença significativa de potência reativa, gerando desperdício de energia. A equação para o cálculo do fator de potência é:

$$FP = \cos(\theta) = \frac{P}{S}, \tag{3.6}$$

onde:

- FP = é o fator de Potência (adimensional)

3.6.1.5 Corrente Elétrica

Corrente Elétrica (I): medida em amperes (A), a corrente elétrica é definida como a quantidade de carga elétrica que atravessa uma seção transversal de um condutor em um intervalo de tempo específico. A fórmula principal para calcular a corrente elétrica é:

$$I = \frac{Q}{t}, \tag{3.7}$$

onde:

- I = Corrente elétrica (A)
- Q = Carga elétrica (Coulombs)
- t = Tempo (segundos)

Uma vez conhecida a tensão eletrica e a resistencia da carga podemos calcular a corrente eletrica pela lei de Ohn :

$$I = \frac{V}{R} \tag{3.8}$$

onde:

- V = Tensão (V)
- R = Resistência elétrica (Ω)

3.6.1.6 Tensão Elétrica

Tensão Elétrica (V): diferença de potencial elétrico entre dois pontos, medida em volts (V). É a grandeza responsável por gerar o fluxo de corrente em um circuito. A relação entre a tensão, a corrente (I) e a resistência (R) é dada pela Lei de Ohm:

$$V = I \times R \tag{3.9}$$

onde:

- V = Tensão (V)
- I = Corrente elétrica (A)
- R = Resistência elétrica (Ω)

3.6.1.7 Tensão RMS

Tensão RMS ($V_{\mathbf{RMS}}$): medida em volts (V), é o valor da tensão eficaz equivalente a uma tensão contínua que produziria a mesma potência média em uma carga resistiva. RMS significa *Root Mean Square* (raiz quadrada da média dos quadrados), sendo especialmente útil para sinais de corrente alternada (CA).

$$V_{\text{RMS}} = \frac{V_{\text{pico}}}{\sqrt{2}} \tag{3.10}$$

onde:

- V_{RMS} = Tensão eficaz (RMS)
- V_{pico} = Tensão de pico

A potência ativa é obtida pela relação:

$$P = V_{\text{RMS}} \times I_{\text{RMS}} \times \cos(\theta) \tag{3.11}$$

onde θ é o ângulo de fase entre tensão e corrente.

3.6.1.8 Energia Total Consumida (E)

A energia consumida representa a quantidade total de energia utilizada em um período, medida em quilowatt-hora (kWh). Pode ser calculada por:

$$E = P \times t \tag{3.12}$$

onde:

- E = Energia total consumida (kWh)
- t = Tempo de operação (horas)
- P = Potência média demandada, em quilowatts (kW);

Esta medida permite a análise do consumo ao longo do tempo, facilitando a identificação de padrões e a otimização do uso de energia.

Neste capítulo, exploramos os conceitos fundamentais relacionados ao *TinyML*, incluindo técnicas de *machine learning*, otimização de modelos e parâmetros elétricos, como tensão e potência. Esses conceitos são essenciais para entender como implementar modelos de aprendizado em dispositivos com recursos limitados, como os usados em sistemas de monitoramento de energia.

Essas informações servirão de base para o modelo proposto no próximo capítulo, onde detalharemos a metodologia que será utilizada para aplicar o *TinyML* na previsão de consumo energético e detecção de anomalias, com foco em dispositivos IoT.

4 Metodologia

Neste capítulo, é apresentada a metodologia utilizada na construção e desenvolvimento do protótipo. Serão descritos os materiais e os componentes eletrônicos usados, com suas principais características e funções no sistema. Além disso, será explicado como o protótipo foi montado, incluindo a integração dos componentes e as técnicas adotadas para garantir o correto funcionamento e a precisão das medidas obtidas. Também será abordado o desenvolvimento do software, com destaque para as partes do código-fonte que implementam as funcionalidades da placa ESP-32 e a criação do servidor web responsável por coletar, processar e exibir os dados.

4.1 Etapas para Elaboração do Protótipo

Para o desenvolvimento do protótipo de monitoramento de energia, o processo foi organizado em etapas sequenciais, com o objetivo de garantir precisão e eficácia. As etapas desenvolvidas neste trabalho são:

1. **Definição dos Materiais:** Seleção e aquisição dos componentes necessários, como sensores de energia, microcontroladores e dispositivos de comunicação. A escolha dos materiais é essencial para garantir a compatibilidade e o desempenho adequado do protótipo.

2. **Definição da Arquitetura do Sistema:** Estabelecimento da interação entre os componentes, com a definição da configuração de hardware e da estrutura de software. A arquitetura deve ser projetada para facilitar a integração dos componentes e a expansão futura do sistema.

3. **Ajuste e Calibragem dos Materiais:** Realização de ajustes finos e calibração dos sensores e componentes para garantir a precisão das medições. Esta etapa visa reduzir erros nos dados coletados.

4. **Definição do Cenário Real de Aplicação:** Especificação do ambiente e das condições em que o protótipo será utilizado. Isso inclui a definição dos parâmetros de operação, como a faixa de consumo de energia a ser monitorada, e a adaptação do protótipo às condições ambientais do local.

5. **Escolha e Desenvolvimento dos Modelos de Algoritmos:** Seleção e programação de algoritmos para monitoramento contínuo, detecção de anomalias e predição do consumo de energia. Deve-se escolher modelos de Machine Learning ou algoritmos estatísticos adequados para analisar os dados coletados e fornecer *insights*.

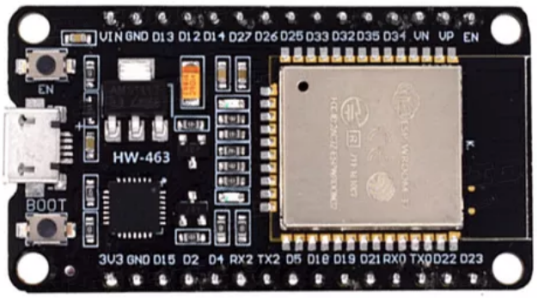
6. **Implementação e Testes:** Montagem do protótipo e programação dos algoritmos, seguidos de testes rigorosos para validar todas as funcionalidades. Testes no cenário real ajudam a identificar falhas ou a necessidade de ajustes antes do lançamento final.

4.2 Descrição dos Materiais Utilizados

4.2.1 Microcontrolador ESP32

Lançado em 2016 pela Espressif Systems, o ESP32, como mostrado na Figura 11, destaca-se como uma solução de baixo custo e alto desempenho para projetos tecnológicos. Amplamente utilizado em aplicações de IoT, ele integra um processador dual-core de 32 bits baseado na arquitetura Xtensa LX6, capaz de operar até 240 MHz, além de possuir conectividade Wi-Fi (802.11 b/g/n) e Bluetooth (clássico e BLE). Projetado para eficiência energética, o microcontrolador oferece modos como o deep sleep, ideais para dispositivos alimentados por bateria ou em ambientes que exigem longa autonomia [114].

Figura 11 – ESP32



Fonte: Figura extraída de [115]

No contexto deste trabalho, o ESP32 atua como unidade central de processamento, responsável pela coleta de dados provenientes de sensores de corrente e tensão, bem como pela execução de modelos de *TinyML* para detecção de padrões de consumo. Sua capacidade de processamento local reduz significativamente a latência e elimina a dependência de comunicação com servidores externos. Entre os principais motivos para sua escolha estão o custo acessível, o baixo consumo energético, o suporte nativo a conectividade **Wi-Fi** e sua adequação para implementações de *TinyML*. As características técnicas mais relevantes do ESP32 estão resumidas na 5.

Tabela 5 – Características técnicas do ESP32

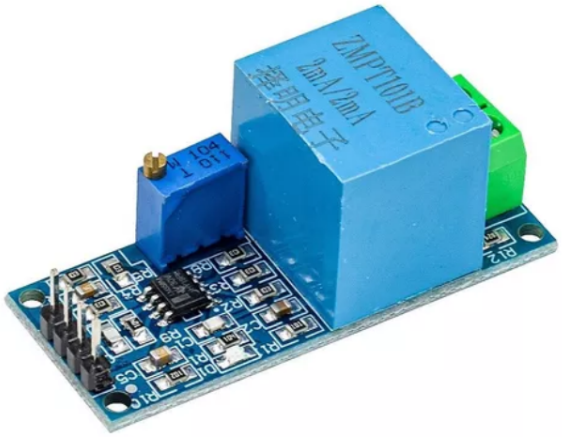
Característica	**Descrição**
Processador	Dual-core Xtensa LX6
Memória SRAM	520 KB
Memória Flash Externa	4 MB a 16 MB
Conectividade	**Wi-Fi**, **Bluetooth**
Periféricos	GPIO, ADC, DAC
Número de GPIOs	34
Pinos Analógicos/Digitais	18 pinos
Tensão de Operação	3,3 V

Fonte: Adaptado do Datasheet [116]

4.3 Sensor De Tensão ZMPT101B

O módulo ZMPT101B destaca-se como uma solução compacta e precisa para medições de tensão alternada, sendo amplamente utilizado em sistemas embarcados, como Arduino e ESP32. Reconhecido por sua alta confiabilidade e estabilidade, o sensor é ideal para monitoramento de sistemas elétricos residenciais e industriais, suportando tensões de até 250 V [117]. Seu design cuidadoso minimiza interferências externas, assegurando leituras consistentes e precisas, como mostra na figura.

Figura 12 – Módulo Sensor ZMPT101B



Fonte: Figura extraída de [117]

Uma característica notável do ZMPT101B é o potenciômetro multivoltas, que permite ajustes finos na saída analógica, facilitando a calibração conforme as necessidades específicas da aplicação. Esse recurso garante maior precisão nos dados capturados, tornando-o adequado para cálculos de potência elétrica em tempo real. Na tabela 6 é apresentada suas principais especificações técnicas, destacando sua robustez e versatilidade. O sensor foi escolhido devido à sua facilidade de integração, baixo custo e excelente

desempenho em medições residenciais.

Tabela 6 – Especificações do Sensor Zmpt101B

Característica	**Descrição**
Tensão de alimentação	5 a 30 VDC
Faixa de medição	0 a 250 VAC
Corrente de entrada nominal	2 mA
Corrente de saída nominal	2 mA
Linearidade	0,2%
Isolamento elétrico	4000 V
Precisão	±0,5%
Temperatura de operação	-40ºC a 70ºC
Dimensões	22 mm ×20 mm × 51 mm

Fonte: Adptado de [117]

4.3.1 Sensor de Corrente SCT-013

O SCT-013 é um sensor de corrente não invasivo do tipo alicate amperímetro, projetado para medir correntes alternadas de até 100 A. Trata-se de um transformador de corrente ou CT (*Current Transformer*) amplamente utilizado no monitoramento de instalações elétricas. Sua principal vantagem é a facilidade de uso, pois pode ser instalado diretamente no cabo condutor de fase ou neutro sem interromper o circuito ou expor o usuário a altas tensões. O design split-core (núcleo dividido) do SCT-013 permite que o sensor "abrace"o cabo a ser monitorado, eliminando a necessidade de desconectar ou modificar o circuito. Internamente, uma bobina realiza a medição da corrente elétrica, conforme ilustrado na figura 13. Disponível em diversos modelos, o SCT-013 atende diferentes cenários de aplicação, desde monitoramento residencial até sistemas industriais. Cada versão é adaptada para atender requisitos específicos, garantindo flexibilidade e precisão nas medições.[118].

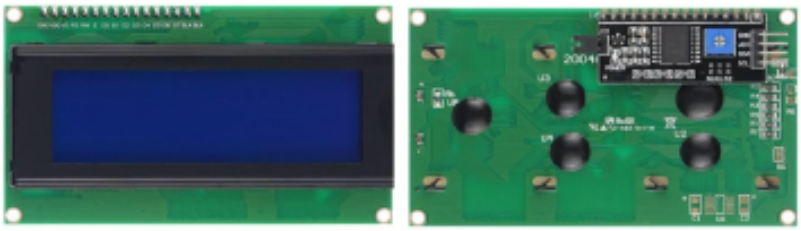
Figura 13 – Sensor de Corrente SCT-013-000

Fonte: Figura extraída de [118]

4.3.2 Display LCD 20x4

O display LCD (*Liquid Crystal Display*) 20x4 é uma solução eficiente para exibir informações em tempo real, sendo ideal para o monitoramento de dados coletados pelos sensores do projeto. Utilizando tecnologia de cristal líquido, ele consome apenas cerca de 1 mA quando a luz de fundo está desligada, garantindo baixo consumo energético. Possui a capacidade para exibir até 80 caracteres organizados em 4 linhas de 20 colunas, o dispositivo permite visualizar um grande volume de informações de forma compacta e prática [119]. Sua integração ao sistema facilita o acesso imediato aos dados relevantes sobre o consumo de energia, conforme ilustrado na Figura 14.

Figura 14 – Display LCD



Fonte: Figura extraída de [120]

4.3.3 Módulo Relé de 2 Canais

O Módulo Relé de 2 Canais, como ilustrado na Figura 15 é uma solução versátil para automação e controle de dispositivos elétricos, proporcionando uma interface segura entre circuitos de baixa potência e cargas mais elevadas. Compatível com microcontroladores como Arduino e ESP32, ele utiliza relés eletromecânicos controlados por tensão de 5V e oferece isolamento elétrico via optoacopladores, garantindo proteção contra interferências. Cada canal inclui um LED (*Light Emitting Diode*) indicador que sinaliza o estado do relé, facilitando a visualização durante operação. Seus terminais de conexão simplificam a ligação de dispositivos externos, como lâmpadas, motores e eletrodomésticos, que operam em corrente alternada AC (*Alternating Current*) ou contínua DC (*Direct Current*). Robusto e fácil de integrar, o módulo é amplamente utilizado em automação residencial, controle de demanda e aplicações IoT [121].

No contexto deste trabalho, o módulo será acionado com base em dados coletados pelos sensores e previsões geradas pelo sistema, permitindo o controle automatizado de cargas conectadas. Essa funcionalidade é essencial para implementar estratégias de controle de demanda e automação residencial.

Na tabela 7 é apresentado os detalhes técnicos do relé JQC-3FF-S-Z, destacando suas especificações de desempenho e operação.

Figura 15 – Módulo Relé de 2 Canais

Fonte: [122]

Tabela 7 – Especificações do Relé JQC-3FF-S-Z

Parâmetros	**Valores**
Modelo	JQC-3FF-S-Z
Tempo de acionamento de contato	10 ms
Temperatura de funcionamento	-45 a 85 °C
Proteção do relé	IP67
Peso unitário	10 g
Umidade de operação	-40 a 80%
Tensão de operação	5 V
Tensão máxima de comutação	277 VAC / 30 VDC
Corrente máxima de comutação	15 A

Fonte: Datasheet adaptado pelo Autor (2025)

4.3.4 Fonte Ajustável Protoboard 3.3V/5 V

A fonte ajustável protoboard 3.3V/5V, como ilustrada na 16 é um módulo compacto que fornece tensões estáveis de 3.3V e 5V para protoboards, ideal para alimentar microcontroladores como Arduino e ESP32. Aceita entrada de 6.5 a 12 VDC (*Volts Direct Current*) ou via USB e utiliza reguladores LM1117 para garantir segurança e precisão. No projeto, a fonte é fundamental para alimentar os sensores e microcontroladores com tensões adequadas, garantindo o funcionamento estável do sistema Na tabela 8 são detalhadas suas principais especificações técnicas.

Figura 16 – Fonte Ajustável Protoboard 3.3V/5 V

Fonte: Figura extraída de [123]

Tabela 8 – Características da Fonte ajustável Protoboard 3.3V/5V

Característica	**Descrição**
Tensões de saída	3,3V e 5 V ajustáveis
Tensão de entrada	6,5 a 12 VDC ou via cabo USB
Corrente máxima de saída	700 mA
Reguladores de tensão	LM1117
Indicadores	LEDs para status de tensão
Conectores	Compatíveis com pinos e jumper wires
Saídas simultâneas	3,3 V e 5 V em diferentes barramentos
Aplicações	ESP32, Arduino, PIC e MCUs

Fonte: Datasheet adptado pelo autor (2025)

4.3.5 Resistores e Capacitores

Os resistores e capacitores são componentes fundamentais em circuitos eletrônicos, desempenhando papéis distintos, mas complementares. Os resistores, como mostrado na Figura 17 (a), limitam a corrente elétrica em um circuito, dissipando energia na forma de calor [124]. Enquanto os capacitores, ilustrados na Figura 17 (b), armazenam e liberam energia, sendo amplamente utilizados como filtros, temporizadores ou estabilizadores de tensão. A combinação desses componentes é comum em circuitos RC (*Resistor-Capacitor*), onde são empregados para gerar atrasos, filtrar sinais ou estabilizar sistemas [124].

(a) Resistor – Fonte: Figura extraída de [125]

(b) Capacitor – Fonte: Figura extraída de [126]

Figura 17 – Componentes eletrônicos: (a) Resistor, (b) Capacitor.

No contexto deste projeto, dois resistores de 10 kΩ são utilizados como resistores de pull-up/pull-down, garantindo a estabilidade dos sinais digitais no circuito. Além disso, um capacitor de 22 μF atua como filtro de desacoplamento, minimizando ruídos e variações na tensão de alimentação. Essa configuração assegura maior estabilidade para o microcontrolador e outros componentes sensíveis, otimizando o funcionamento do sistema.

4.3.6 Protoboard

A protoboard como ilustrado na 18 é uma ferramenta essencial para o desenvolvimento e teste de circuitos eletrônicos, destacando-se por sua praticidade e versatilidade. Ela permite a montagem temporária de componentes sem a necessidade de solda, facilitando experimentações e ajustes rápidos durante a prototipagem. Sua matriz de furos interconectados possibilita a conexão direta de componentes como resistores, capacitores e microcontroladores, enquanto barramentos dedicados fornecem alimentação elétrica de forma organizada [127].

No projeto, a protoboard foi escolhida devido ao seu baixo custo e facilidade de uso, permitindo a criação e modificação de conexões temporárias com fios jumper. Essa característica é especialmente útil para testes iniciais e ensaios antes da transição para placas de circuito impresso ou em inglês PCB (*Printed Circuit Board*), garantindo um desenvolvimento ágil e econômico.

Figura 18 – Protoboard

Fonte: Figura extraida de [127]

4.4 Detalhamento do custo do Protótipo

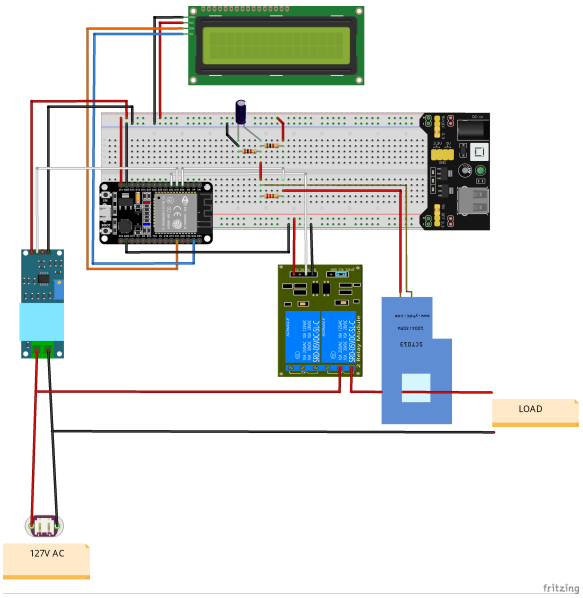
No apêndice B é apresentada a Tabela 13, que detalha os custos estimados para o desenvolvimento do protótipo, incluindo todos os componentes essenciais utilizados na montagem do sistema. A inclusão desse anexo visa reforçar a transparência e a viabilidade econômica do projeto, demonstrando que a solução proposta é acessível e pode ser replicada em diferentes contextos acadêmicos, industriais e residenciais com baixo investimento.

4.5 Modelo do Sistema Proposto

4.5.1 Arquitetura do sistema

A arquitetura do sistema proposto, ilustrada na Figura 19, foi projetada para monitorar e realizar análises preditivas do consumo de energia em ambientes residenciais. O sistema combina tecnologias de Machine Learning Embarcado para oferecer uma solução robusta e eficiente, que otimiza o uso de energia e promove uma gestão inteligente dos recursos energéticos. A modularidade da arquitetura permite que cada componente seja facilmente integrado ou substituído, facilitando futuras expansões e ajustes. Além disso, o design baseado em protoboard simplifica a prototipagem e os testes iniciais, antes da transição para uma placa de circuito impresso definitiva. O sistema é composto por um conjunto de componentes eletrônicos e programáveis interconectados, projetados para capturar dados em tempo real, processá-los e extrair informações significativas para a análise do consumo energético.

Figura 19 – Esquema elétrico do modelo proposto



Fonte: Autor(2025)

4.6 Recursos de *Softwares*

Para o desenvolvimento deste trabalho de análise preditiva em tempo real para *smart meters*, foram empregados diversos recursos tecnológicos e uma infraestrutura de software adaptada à realidade de dispositivos embarcados com recursos computacionais limitados. Esses recursos foram fundamentais para viabilizar o treinamento, a conversão e a execução local dos modelos de *Machine Learning*, assegurando a eficiência energética, a precisão das previsões e a detecção de anomalias em tempo real.

A arquitetura de software foi projetada para atender às demandas de processamento embarcado utilizando *TinyML*, e também para oferecer uma interface de monitoramento acessível e armazenamento de dados eficiente. Os principais recursos empregados incluem:

- **Scikit-learn:** Utilizado para o treinamento dos modelos de aprendizado supervisionado *XGBoost* e não supervisionado *One-Class SVM*, permitindo uma análise robusta dos dados históricos de consumo energético.
- **TensorFlow e Keras:** Aplicados no desenvolvimento e treinamento do modelo *Autoencoder*, voltado à detecção de anomalias mais complexas com base na reconstrução dos padrões de consumo.
- **TensorFlow Lite (*TFLite*):** Utilizado na quantização e *pruning* do *Autoencoder*, permitindo sua execução eficiente em microcontroladores com restrições de memória e processamento.
- **Micromlgen:** Biblioteca empregada para converter os modelos treinados de *XGBoost* e *One-Class SVM* em código C++ compatível com o ambiente embarcado do ESP32.
- **Arduino IDE:** Ambiente principal de desenvolvimento utilizado para programar o microcontrolador ESP32, integrar sensores, gerenciar o fluxo de dados e realizar inferências em tempo real.
- **Arduino OTA (*Over-The-Air*):** Recurso habilitado no ESP32 que permite a atualização remota do firmware por meio da rede *Wi-Fi*. Essa funcionalidade elimina a necessidade de acesso físico ao dispositivo, facilitando ajustes, correções e melhorias no sistema mesmo após sua implantação.
- **MQTT (*Message Queuing Telemetry Transport*):** Protocolo leve de comunicação *publish/subscribe* utilizado para a transmissão eficiente de dados entre o ESP32 e os serviços de monitoramento, ideal para redes com largura de banda limitada.

- **Eclipse Mosquitto:** Servidor *broker* MQTT utilizado para intermediar a comunicação entre os dispositivos e os serviços de backend, garantindo confiabilidade na troca de mensagens em tempo real.
- **MongoDB:** Sistema de banco de dados NoSQL utilizado para armazenar as leituras históricas de consumo energético. Sua estrutura escalável facilita consultas rápidas e a análise longitudinal dos dados.
- **Node-RED:** Plataforma de desenvolvimento baseada em fluxo visual, utilizada para criar *dashboards* interativos em tempo real, recebendo dados do ESP32 e exibindo-os de forma intuitiva para os usuários. O Node-RED também foi utilizado para orquestrar o envio de dados ao MongoDB e emitir alertas em caso de detecção de padrões anômalos.
- **Jupyter Notebook:** Ferramenta utilizada na etapa exploratória para análise estatística, visualização dos padrões de consumo e validação preliminar dos modelos.

Essa infraestrutura híbrida integra o processamento embarcado (*edge computing*) com armazenamento em nuvem e visualização remota, proporcionando um sistema robusto, escalável e responsivo. A combinação dessas ferramentas permite garantir a portabilidade e eficiência do sistema, viabilizando sua aplicação tanto em ambientes residenciais quanto industriais, mesmo em contextos de conectividade limitada ou intermitente. Assim, o sistema atende às demandas de processamento local para respostas rápidas, ao mesmo tempo que assegura armazenamento confiável e monitoramento acessível remotamente.

4.6.1 Testes e Calibração dos Sensores

Durante os testes práticos, foram utilizadas cargas elétricas (resistivas e indutivas) reais para simular diferentes níveis de consumo energético. Esses dispositivos permitiram validar o sistema em diversos cenários operacionais além daquele proposto originalmente neste projeto.

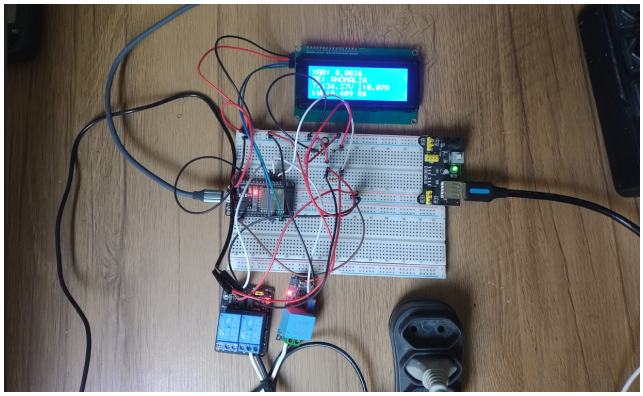
Na tabela 9 é apresentada as cargas utilizadas em bancada para os testes de calibração dos sensores. A utilização dessas cargas foi essencial para ajustar os parâmetros de calibração dos sensores e validar a funcionalidade do protótipo em condições reais de operação. Como pode ser visto na Figura 20, o protótipo em funcionamento apresentou estabilidade e resposta adequada sob diferentes condições de carga.

Tabela 9 – Cargas elétricas testadas (resistivas e indutivas)

Carga	**Potência**	**Corrente**	**Tensão**
Ventilador	150 W	0,96 A 0,90 A 0,87 A	127 V
Umidificador	70 W	0,34 A 0,35 A 0,41 A	127 V
Secador	1700 W	0,52 A 6,72 A 6,98 A 13.97 A	127 V
Liquidificador	500 W	0,97 A 1,32 A	127 V
Geladeira	109 W	1,6 A	127 V

Fonte: Autor (2025)

Figura 20 – Protótipo em funcionamento durante os testes práticos.



Fonte: Autor(2025)

4.6.2 Teste do sensor de tensão ZMPT101B

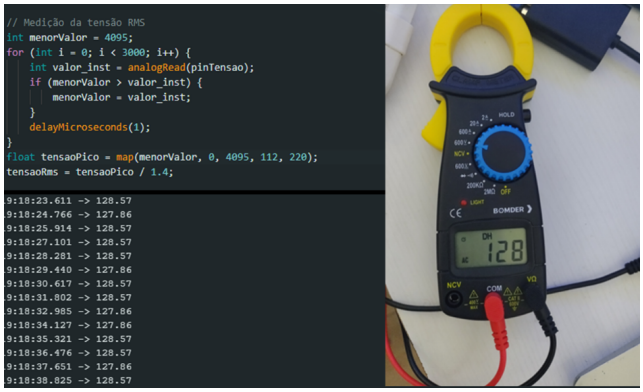
O sensor de tensão ZMPT101B é um módulo de alta precisão projetado para detectar a presença de tensão alternada em circuitos elétricos ou realizar medições de valores de tensão. De acordo com o datasheet do dispositivo [117], o sensor possui um *offset* fixo de 2,5 V em sua saída analógica, que deve ser considerado ao processar os sinais obtidos.

Para avaliar o desempenho do sensor ZMPT101B, foi realizado um experimento utilizando um potenciômetro conectado ao microcontrolador ESP32. O conversor analógico-digital ou em inglês ADC (*Analog-to-Digital Converter*) do ESP32 possui resolução de 12 bits e opera em uma faixa de tensão entre 0 e 3,3 V, o que permite representar até $2^{12} = 4096$ níveis discretos de tensão, conforme especificado no manual técnico do dispo-

sitivo [114]. Durante o experimento, os valores brutos obtidos pelo ADC foram convertidos para as tensões reais da rede elétrica utilizando a função `map()` presente no ambiente de desenvolvimento Arduino IDE. Essa função realiza o escalonamento linear dos valores lidos, ajustando-os para a faixa de tensão esperada e levando em consideração o *offset* de 2,5 V fornecido pelo sensor, conforme descrito na documentação oficial da plataforma Arduino [128]

Os resultados obtidos no teste são apresentados na Figura 21, que compara as tensões reais medidas na rede elétrica com as tensões calculadas pelo código implementado. Observou-se que o ajuste adequado do offset e a calibração cuidadosa do software foram fundamentais para garantir a precisão das medições. Além disso, o uso do ESP32 proporcionou flexibilidade no processamento dos dados e facilitou a integração com outros sistemas embarcados. A diferença entre a tensão calculada e a tensão real foi de aproximadamente 0,70 V, o que representa um erro percentual de cerca de 0,55%, indicando um bom nível de precisão na estimativa da tensão RMS pelo sistema desenvolvido.

Figura 21 – Comparação entre o valor real e o valor obtido pelo sensor ZMPT101B após a calibração



Fonte: Autor(2025)

4.6.3 Teste do sensor SCT-013

O sensor SCT-013 é um transformador de corrente não invasivo utilizado para medir corrente alternada, sendo ideal para aplicações de monitoramento de consumo energético. Ele opera por meio da indução eletromagnética, convertendo a corrente primária que passa pelo condutor em uma corrente proporcional no enrolamento secundário.

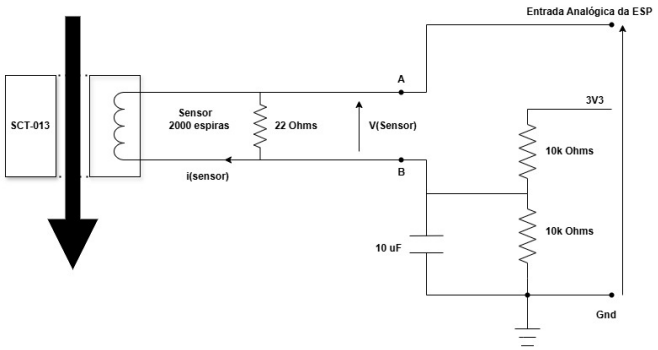
O modelo empregado no protótipo possui uma relação de espiras de 2000:1 e é capaz de medir até 100 A_{RMS}. A saída do sensor é uma corrente alternada proporcional à corrente medida no condutor primário. Para interfaceá-lo ao conversor analógico-digital do ESP32, é necessário converter essa corrente em tensão utilizando um resistor de carga. Além disso, foi implementado um circuito de *offset*, garantindo que o sinal fique dentro

da faixa de entrada do ADC (0–3,3 V), centrado em 1,65 V (VCC/2). Para determinar o valor adequado do resistor de carga, foram realizados os seguintes passos [129]:

1. **Determinar a corrente RMS máxima:** $I_{RMS} = 100$ A.
2. **Converter a corrente RMS para pico:** $I_{pico} = \sqrt{2} \cdot I_{RMS} = 1{,}414 \cdot 100 = 141{,}4$ A.
3. **Corrente de pico no secundário do SCT-013:** $I_{sensor} = \frac{141{,}4}{2000} = 0{,}0707$ A.
4. **Cálculo do resistor de carga:** $R_{carga} = \frac{1{,}65 \text{ V}}{0{,}0707 \text{ A}} \approx 23{,}35\ \Omega$.

Posteriormente, selecionou - se um resistor comercial de 22 Ω, valor próximo ao calculado, como mostrado na Figura 22. Nessa figura, pode-se observar o circuito de condicionamento de sinal responsável por adaptar a saída do sensor SCT-013 para leitura pela entrada analógica do ESP32. O resistor converte a corrente induzida em tensão, enquanto o divisor resistivo e o capacitor formam um circuito de *offset* e filtragem, centralizando o sinal em 1,65 V, dentro da faixa operacional do ADC.

Figura 22 – Circuito para o sensor SCT-013 conectado ao ESP32.

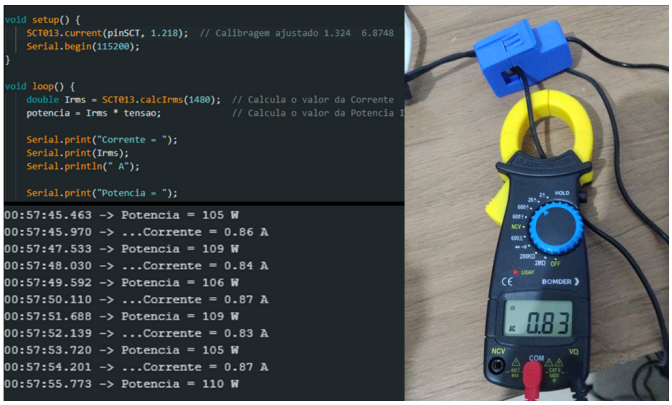


Fonte: Figura extraida de [129]

A pequena diferença entre o valor teórico e o resistor comercial foi compensada durante a calibração do sistema. Após o dimensionamento do resistor e a montagem do circuito de *offset*, as medições obtidas foram comparadas com os valores fornecidos por um alicate amperímetro, como referência, conforme ilustrado na Figura 23.

Os resultados demonstraram que o sensor SCT-013 apresenta boa precisão após a calibração, validando sua aplicação em sistemas de monitoramento de corrente alternada. A diferença percentual entre o valor real medido pelo alicate amperímetro e o valor calculado pelo sensor foi de aproximadamente 4,8%, o que corresponde a uma diferença absoluta de cerca de 0,04 A, indicando uma margem de erro aceitável para aplicações residenciais e protótipos de baixo custo. As cargas elétricas utilizadas permitiram testar o sistema em diferentes níveis de consumo, contribuindo para a confiabilidade do protótipo em cenários reais de operação.

Figura 23 – Comparação entre valores reais e os obtidos pelo SCT-013 após calibração

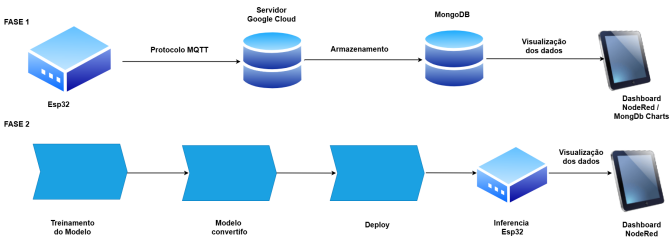


Fonte: Autor(2025)

4.7 Processo de Coleta e Pré-Processamento dos Dados

O processo de coleta, análise e previsão do consumo de energia foi estruturado em uma arquitetura baseada em *Internet das Coisas*, utilizando o microcontrolador ESP32, o protocolo *MQTT* [130], serviços de armazenamento em nuvem (*MongoDB*) [131] e técnicas de aprendizado de máquina embarcado. O fluxo completo é dividido em duas fases principais: coleta e armazenamento e treinamento e inferência, conforme ilustrado na Figura 24.

Figura 24 – Arquitetura para coleta e monitoramento de dados



Fonte: Autor (2025)

Fase 1 – Coleta e Armazenamento: Nesta etapa, o microcontrolador ESP32 realiza medições contínuas de parâmetros elétricos, como tensão, corrente e potência, em tempo real. Os dados capturados são transmitidos via protocolo *MQTT* para um servidor hospedado na *Google Cloud*, garantindo comunicação eficiente e de baixa latência. Uma vez recebidos, os dados são armazenados em um banco de dados NoSQL *MongoDB*, que oferece flexibilidade para lidar com grandes volumes de informações não estruturadas. As informações armazenadas podem ser acessadas e visualizadas em dashboards interativos desenvolvidos com *Node-RED* ou *MongoDB Charts*, permitindo um monitoramento intuitivo e em tempo real do consumo energético.

Fase 2 – Treinamento e Inferência: Com os dados previamente coletados, é

realizado o treinamento de modelos de aprendizado de máquina, os quais são otimizados para tarefas como previsão de consumo, detecção de anomalias ou identificação de padrões relevantes. Após o treinamento, o modelo é convertido para um formato compatível com dispositivos embarcados, como o ESP32, utilizando frameworks específicos (por exemplo, *TensorFlow Lite*). Na etapa de inferência, o ESP32 passa a executar localmente as previsões em tempo real, sem a necessidade de enviar dados continuamente para a nuvem, o que reduz custos operacionais e aumenta a eficiência. Os resultados gerados pelo modelo embarcado são então enviados para o dashboard do *Node-RED*, proporcionando uma visualização clara e rápida para tomada de decisões.

Esse fluxo integrado entre coleta, armazenamento, aprendizado e inferência embarcada representa uma solução robusta para sistemas de monitoramento inteligente, combinando IoT e inteligência artificial para promover eficiência energética e respostas proativas frente a variações no consumo.

4.7.1 Coleta de Dados

Durante 31 dias, foram registradas medições contínuas do consumo energético de uma geladeira, possibilitando uma análise preditiva detalhada. Os principais parâmetros capturados durante o experimento incluem:

- **Tensão (V):** Medida continuamente em tempo real para monitorar variações na rede elétrica e identificar possíveis irregularidades.
- **Corrente (A):** Monitorada simultaneamente à tensão, permitindo o cálculo do consumo instantâneo da geladeira.
- **Potência (W):** Calculada a partir da tensão e corrente, com gráficos de barras diários que revelaram padrões de uso ao longo do período.
- **Consumo em kWh:** Registrado e acumulado diariamente, exibido em gráficos para análise de tendências e identificação de picos de consumo.
- **Custo total (R$):** Calculado com base no consumo acumulado e nas tarifas vigentes, facilitando o acompanhamento financeiro.

A coleta de dados inicia-se com o microcontrolador, responsável por capturar informações dos sensores conectados em tempo real. Após a coleta, os dados são transmitidos para um servidor na nuvem *Google Cloud Server* utilizando o protocolo MQTT. O servidor atua como intermediário, armazenando temporariamente as informações e gerenciando seu fluxo até o próximo estágio.

Os dados transmitidos são armazenados em um banco de dados NoSQL, o MongoDB, onde são organizados no formato *JSON-like*. Esse formato permite consultas eficientes e facilita o processamento das informações. Para análise detalhada, os dados foram exportados para um arquivo CSV e carregados em um DataFrame utilizando a biblioteca Pandas. O índice do DataFrame foi convertido para o formato **datetime**, permitindo operações precisas relacionadas ao tempo.

Com base no *timestamp*, foram criadas duas novas colunas:

- **'data'**, contendo apenas a data (dia do calendário);

- **'hora'**, contendo apenas a hora do dia.

Além disso, foi adicionada uma coluna chamada **'dia_da_semana'**, que representa o dia da semana como um número inteiro, variando de 0 (segunda-feira) a 6 (domingo). Essa informação foi útil para identificar padrões de consumo relacionados aos dias úteis e aos finais de semana. Para simplificar o conjunto de dados e remover redundâncias, colunas desnecessárias ou cumulativas, como *sensor/kWhTotal*, *sensor/custo*, *sensor/potência* e *_id*, foram excluídas. Em seguida, os dados foram agrupados por **'data'** e **'hora'**, calculando-se a média das medições horárias. Essa etapa foi essencial para organizar as informações de forma mais clara e facilitar análises subsequentes.

O processamento estruturado dos dados permitiu a identificação de tendências de consumo energético ao longo do período de 31 dias. Além disso, a organização das informações em *DataFrames* possibilitou a criação de gráficos e visualizações que auxiliaram na extração de insights relevantes sobre o comportamento energético da geladeira.

4.8 Treinamento e Avaliação dos Modelos

Após o pré-processamento, os dados foram utilizados para treinar modelos de aprendizado de máquina capazes de prever o consumo diário de energia (kWh) e detectar fraudes ou anomalias no consumo. Para garantir robustez e versatilidade, três algoritmos principais foram testado para implementação: *XGBoost regressor*, *One - class SVM* e *Autoencoders*. Cada um desses algoritmos foi escolhido por suas características específicas e aplicabilidade ao conforme discutido anteriormente. Os dados foram divididos em duas partes:

- Conjunto de treinamento (70%) : Usado para treinar os modelos.
- Conjunto de teste (30%) : Usado para avaliar o desempenho dos modelos com dados que eles nunca viram antes.

4.8.1 Escolha dos Modelos e Ajuste de Hiperparâmetros

Os três modelos foram testados para abordar diferentes aspectos do problema. O *XGBoost Regressor* foi escolhido para prever o consumo diário de energia. Este algoritmo baseado em árvores de decisão combina múltiplas árvores para melhorar a precisão das previsões. Ele foi configurado para utilizar a GPU, acelerando o processo de treinamento. Para encontrar os melhores valores de configuração do modelo, foi usada a técnica chamada *Grid Search*. Nessa abordagem, várias combinações de hiperparâmetros são testadas para identificar aquela que proporciona o menor erro.

One-Class SVM é um algoritmo de aprendizado de máquina usado para detecção de anomalias ou outliers. Diferentemente do *SVM* tradicional, que classifica dados em duas ou mais classes, o *One-Class SVM* aprende a fronteira que delimita os dados "normais"e identifica qualquer ponto fora dessa fronteira como uma anomalia.

Além do *SVM*, um modelo de *Autoencoder* foi treinado para detectar outliers. *Autoencoders* são redes neurais projetadas para aprender uma representação comprimida dos dados normais. Se um dado apresentar uma reconstrução ruim, ele é considerado uma anomalia. Os hiperparâmetros utilizados para cada algoritmo são apresentados na tabela.

Tabela 10 – Hiperparâmetros dos Modelos *TinyML* escolhidos

Modelo	**Hiperparâmetro**	**Valor Ajustado**
XGBoost	max_depth	*Grid Search*
	learning_rate	*Grid Search*
	n_estimators	*Grid Search*
	subsample	*Grid Search*
One-Class SVM	*nu*	0.01
	kernel	*rbf*
	gamma	0.001
	shrinking	*False*
	verbose	*True*
Autoencoder	Neurônios	4
	Ativação	*relu*
	Neurônios Saída	2
	Ativação Saída	*sigmoid*
	Loss Function	*mse*
	Optimizer	*adam*

Fonte: Autor (2025)

4.8.2 Métricas de Avaliação de Desempenho

Após a otimização dos modelos, a avaliação foi realizada utilizando o conjunto de teste (30%), garantindo que os modelos fossem testados com dados que não foram vistos durante o treinamento. O desempenho foi analisado considerando tanto a precisão na

previsão do consumo de energia quanto a eficácia na detecção de anomalias. As métricas de avaliação foram divididas em duas categorias principais:

- **Precisão da Previsão (*XGBoost*):** Mede a capacidade do modelo de prever corretamente o consumo diário de energia.
- **Detecção de Anomalias (*One-Class SVM* e *Autoencoder*):** Avalia a capacidade dos modelos de identificar padrões incomuns no consumo.

Este trabalho adota um conjunto de medidas derivadas para avaliar de forma robusta o desempenho dos modelos empregados. Essas métricas, conforme descrito em [132], são amplamente utilizadas na avaliação de modelos preditivos e sistemas de detecção de anomalias, permitindo análises comparativas consistentes e objetivas. A precisão do modelo *XGBoost* foi avaliada utilizando quatro métricas comuns em regressão:

- **Erro Quadrático Médio (MSE)**: Mede o quão distante as previsões estão dos valores reais, penalizando erros maiores de forma mais significativa, conforme a seguinte expressão.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.1}$$

- **Erro Quadrático Médio da Raiz, ou do inglês *RMSE***: Representa a raiz quadrada do MSE, trazendo a métrica para a mesma unidade dos valores originais, facilitando a interpretação, conforme a seguinte expressão.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2}$$

- **Coeficiente de Determinação (R^2)**: Indica a proporção da variância dos dados que é explicada pelo modelo. Valores próximos de 1 indicam melhor ajuste, conforme a seguinte expressão.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.3}$$

  Onde $\bar{y}$ é a média dos valores observados.

Para avaliar a capacidade dos modelos *one-class SVM* e *Autoencoder* em identificar anomalias, foram utilizadas métricas específicas de detecção de anomalias:

- **Acurácia, ou do inglês *Accuracy*:** Mede a proporção de classificações corretas (tanto anomalias quanto normais) em relação ao total de amostras. Embora seja uma métrica intuitiva, a acurácia pode ser enganosa em conjuntos de dados altamente desbalanceados, pois o modelo pode classificar a maioria das amostras como normais e ainda assim apresentar alta acurácia, conforme a seguinte equação:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4.4}$$

  Onde TP (*True Positive*) é o número de anomalias corretamente identificadas, TN (*True Negative*) é o número de exemplos normais corretamente classificados, FP (*False Positive*) é o número de normais incorretamente classificados como anomalias, e FN (*False Negative*) é o número de anomalias não detectadas.

- **Precisão (*Precision*):** Mede a proporção de anomalias identificadas corretamente em relação ao total de detecções, conforme a seguinte expressão:

$$Precision = \frac{TP}{TP + FP}. \tag{4.5}$$

- **F1-Score:** Representa uma média harmônica entre precisão e revocação, sendo útil para avaliar o equilíbrio entre essas métricas, conforme a seguinte expressão:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{4.6}$$

Por fim, a aplicação dessas métricas permitiu comparar os modelos e identificar a abordagem mais eficiente para a previsão e detecção de anomalias. Além disso, a eficiência computacional dos modelos foi avaliada, garantindo sua viabilidade para execução embarcada no ESP32, onde os recursos computacionais são limitados.

4.8.3 Visualização dos Dados

Para um monitoramento eficiente do consumo energético e a identificação de padrões ou anomalias, o sistema adota uma arquitetura em duas fases que integra análise histórica baseada em nuvem e inferência local embarcada, garantindo precisão, eficiência e privacidade.

Na primeira fase, os dados coletados são armazenados no banco MongoDB e analisados por meio de gráficos interativos utilizando o *MongoDB Charts*. Essa ferramenta oferece dashboards detalhados para monitorar o consumo ao longo do tempo e identificar tendências e padrões. Contudo, devido ao tempo de inserção e processamento, há um atraso médio de aproximadamente 1 minuto na atualização das informações. O armazenamento dos dados históricos também permite o re-treinamento dos modelos de aprendizado

de máquina e a realização de análises retrospectivas para aprimorar a detecção de anomalias.

Na segunda fase, para superar a latência da nuvem e possibilitar a análise em tempo real, foi implementado um dashboard em *Node-RED* que recebe dados processados localmente no microcontrolador ESP32. Este utiliza modelos *TinyML* embarcados para realizar inferência diretamente no dispositivo, detectando variações anômalas instantaneamente sem depender da conectividade com a nuvem. Essa abordagem assegura maior autonomia, privacidade e agilidade na análise dos dados, eliminando a latência típica de soluções baseadas exclusivamente na nuvem.

Na Figura 25 é ilustrado um exemplo de detecção de *outliers* no consumo energético utilizando MongoDB Charts. O painel exibe oscilações incomuns que facilitam a identificação de possíveis falhas ou irregularidades nos equipamentos monitorados. Os principais recursos implementados incluem:

- **Análise histórica com *MongoDB Charts:*** Visualização interativa dos dados armazenados, permitindo a exploração de tendências e suportando o re-treinamento dos modelos com base em dados históricos.
- **Monitoramento em tempo real com *Node-RED:*** Exibição imediata das leituras dos sensores e alertas automáticos para anomalias, viabilizando intervenções rápidas.
- **Processamento local com *TinyML:*** Execução de modelos de aprendizado diretamente no ESP32, reduzindo latência e garantindo maior privacidade e autonomia no processamento.

Figura 25 – Dashboard de Monitoramento utilizando MongoDB Charts

Fonte: Autor(2025)

Essa combinação de ferramentas integra as vantagens da análise aprofundada dos dados históricos com a rapidez e eficiência do processamento embarcado, alinhada à arquitetura de duas fases adotada neste trabalho, garantindo assim um sistema robusto para o gerenciamento do consumo energético.

4.9 Implementação TinyML

A implementação do *TinyML* no ESP32 permitiu a execução local de modelos de *machine learning* eliminando a necessidade de comunicação com servidores externos e, assim, reduzindo latência e consumo de energia. Os modelos *One-Class SVM* e *XGBoost* foram convertidos para C via *m2gen (m2c)*, otimizando projeções e cálculos diretamente no microcontrolador. Para o *Autoencoder*, treinado em *TensorFlow/Keras*, foram aplicadas duas técnicas de otimização cruciais: pruning e conversão para *TensorFlow Lite*.Como mostrado na Figura 26. Primeiramente, o pruning foi utilizado para reduzir o tamanho do modelo, removendo os pesos menos significativos durante o treinamento.

Figura 26 – Aplicação do pruning no Autoencoder

```
# Definir a função de pruning
pruning_params = {
    "pruning_schedule": tfmot.sparsity.keras.PolynomialDecay(
        initial_sparsity=0.30, final_sparsity=0.80,
        begin_step=0, end_step=1000
    )
}

# Criar um modelo idêntico ao original, mas aplicando pruning às camadas Dense
pruned_model = Sequential([
    tfmot.sparsity.keras.prune_low_magnitude(Dense(2, activation='elu', input_shape=(2,)), **pruning_params),
    tfmot.sparsity.keras.prune_low_magnitude(Dense(4, activation='elu'), **pruning_params),
    tfmot.sparsity.keras.prune_low_magnitude(Dense(8, activation='elu'), **pruning_params),
    tfmot.sparsity.keras.prune_low_magnitude(Dense(16, activation='elu'), **pruning_params),
    tfmot.sparsity.keras.prune_low_magnitude(Dense(8, activation='elu'), **pruning_params),
    tfmot.sparsity.keras.prune_low_magnitude(Dense(4, activation='elu'), **pruning_params),
    tfmot.sparsity.keras.prune_low_magnitude(Dense(2, activation='elu'), **pruning_params)
])

# Compilar o modelo novamente
pruned_model.compile(optimizer='adam', loss='mse')
```

Fonte: Autor (2025)

A técnica de pruning de magnitude baixa foi aplicada nas camadas do Autoencoder, removendo pesos com valores próximos de zero, que têm um impacto mínimo no desempenho do modelo, o processo de pruning foi configurado para remover 30% dos pesos no início do treinamento, aumentando progressivamente até 80% ao longo de 1000 etapas de treinamento. Essa otimização resultou em um modelo mais compacto e eficiente, com menor uso de memória e maior eficiência computacional, tornando-o mais adequado para ser executado em dispositivos embarcados como o ESP32. Após a aplicação do pruning, o modelo foi convertido para o formato *TensorFlow Lite* (TFLite) para execução em dispositivos com recursos limitados. A conversão para TFLite é fundamental para garantir que o modelo seja mais rápido, eficiente em termos de memória e menor em tamanho. A conversão foi realizada utilizando o *TFLiteConverter*, como ilustrado na Figura 27, e o modelo resultante foi salvo no formato .tflite.

Ao ser convertido para TFLite, o modelo não apenas ficou mais leve, mas também obteve um desempenho significativamente melhor em inferências rápidas e de baixo custo computacional, ideal para uso em dispositivos de baixo consumo como o ESP32.

Figura 27 – Conversão do modelo para TensorFlow Lite (TFLite)

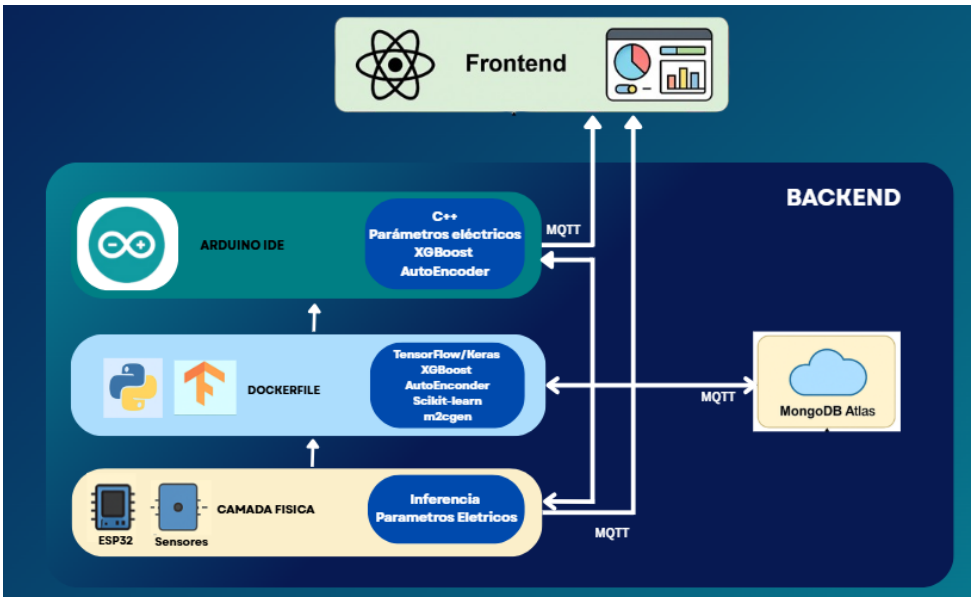
```
# Converter para TensorFlow Lite
converter = tf.lite.TFLiteConverter.from_keras_model(autoencoder)
tflite_model = converter.convert()
open("autoencoder_outlier.tflite", "wb").write(tflite_model)
```

Fonte: Autor (2025)

A avaliação dos três modelos *One-Class SVM*, *XGBoost* e *Autoencoder* considerou precisão, uso de memória e tempo de inferência, permitindo identificar o modelo mais eficiente para a detecção de anomalias no *SP32*. O processo garantiu um equilíbrio ideal entre desempenho e consumo de recursos, permitindo que o Autoencoder, otimizado com pruning e TFLite, fosse a escolha mais eficaz para esse tipo de aplicação.

A arquitetura final do sistema está dividida em duas camadas principais, como é mostrado na Figura 28, o Backend e Frontend.

Figura 28 – Arquitectura do modelo



Fonte: Autor (2025)

O Backend é composto por três módulos interconectados:

- **Camada Física:** Inclui o *ESP32* e os sensores, que coletam parâmetros elétricos. O ESP32 realiza inferência local (*TinyML*) e envia os dados via *MQTT*.
- **Docker Container (Python):** Este módulo, encapsulado por um *Dockerfile*, contém os modelos de Machine Learning (*TensorFlow/Keras, XGBoost, AutoEncoder, Scikit-learn, m2ogen*). Ele recebe dados via *MQTT* e os processa para detecção de anomalias e previsão. Os dados são armazenados no *MongoDB Atlas*.

- **Arduino IDE (C++):** Este módulo é responsável por fazer o deploy do código (parâmetros elétricos, XGBoost, AutoEncoder) para o ESP32.

O Frontend, desenvolvido em *React*, oferece uma interface gráfica para visualização em tempo real dos dados coletados e dos resultados de inferência, comunicando-se com o Backend via *MQTT*.

Essa arquitetura permite um sistema eficiente de monitoramento e previsão de consumo de energia, com detecção de anomalias em tempo real, utilizando as técnicas de *TinyML* para operação em dispositivos de baixo consumo como o *ESP32*.

Para encerrar este capítulo, pode-se afirmar que a metodologia adotada proporcionou uma base sólida para o desenvolvimento de um sistema embarcado inteligente, capaz de realizar medições elétricas, prever o consumo energético e detectar padrões anômalos com eficiência. A integração entre hardware e software, aliada ao uso de técnicas de TinyML, permitiu a criação de um protótipo funcional, de baixo custo e com potencial de aplicação real. No próximo capítulo, serão apresentados e analisados os resultados obtidos a partir dos testes práticos, destacando o desempenho dos modelos implementados e a eficácia da solução proposta

5 Resultados e Discussão

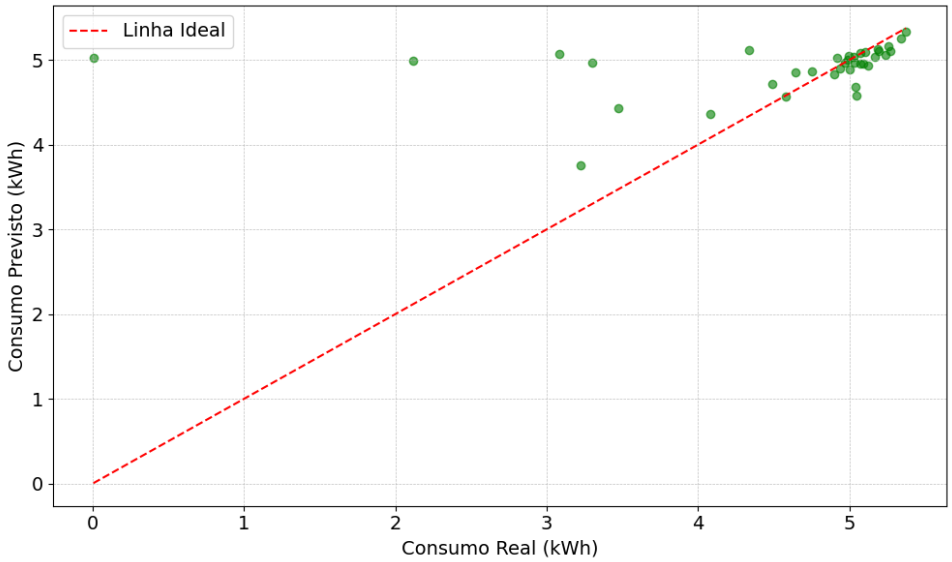
A análise dos resultados avalia a eficácia do modelo *XGBoost* na predição do consumo de energia da geladeira, identificando padrões e possíveis limitações. Diferentes métricas e representações gráficas ajudam a compreender a precisão das previsões, a distribuição dos erros e variações sazonais no consumo.

A detecção de anomalias permitiu identificar padrões anômalos, enquanto a análise dos erros absolutos e residuais ajudou a avaliar a robustez do modelo e os fatores que impactam discrepâncias entre valores reais e previstos. Os resultados oferecem insights valiosos sobre o desempenho do modelo e possíveis melhorias para aumentar a precisão das previsões.

5.1 Análise dos Resultados do XGBoost para Predição do Consumo da Geladeira

Os gráficos apresentados oferecem uma visão detalhada do desempenho do modelo XGBoost na previsão do consumo de energia da geladeira. Na figura 29 é mostrado um gráfico de dispersão que compara os valores reais de consumo (eixo X) com os valores previstos pelo modelo de machine learning (eixo Y).

Figura 29 – Consumo Real vs Previsto



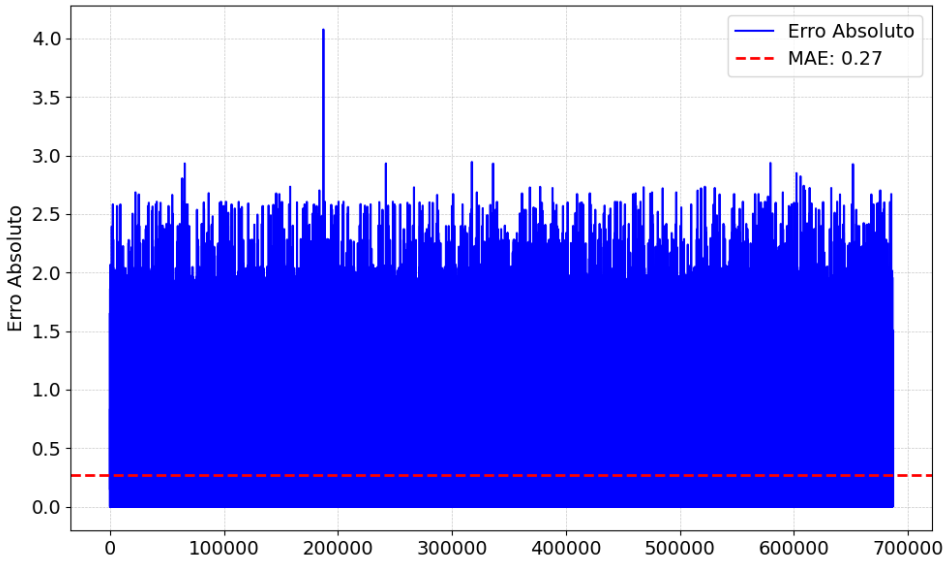
Fonte: Autor(2025)

O gráfico de dispersão mostra a relação entre os dados observados e as previsões geradas pelo modelo *XGBoost*. A linha vermelha tracejada representa o cenário ideal de

previsão perfeita, ou seja, quando os valores previstos são exatamente iguais aos valores reais. A maioria dos pontos está concentrada próximo a essa linha, evidenciando uma boa correlação entre os resultados preditos e os dados reais. Apesar dessa proximidade geral, é possível identificar algumas discrepâncias, especialmente em faixas de consumo mais baixas, onde pequenas variações nos valores reais resultam em desvios mais significativos nas previsões. Esses erros podem estar associados a fatores externos não contemplados no treinamento do modelo, como oscilações na rede elétrica, alterações na temperatura ambiente ou mudanças no padrão de uso do equipamento.

O gráfico do erro absoluto ao longo das amostras na Figura 30 evidencia a magnitude dos desvios entre os valores reais e previstos.

Figura 30 – Erro absoluto ao longo das amostras

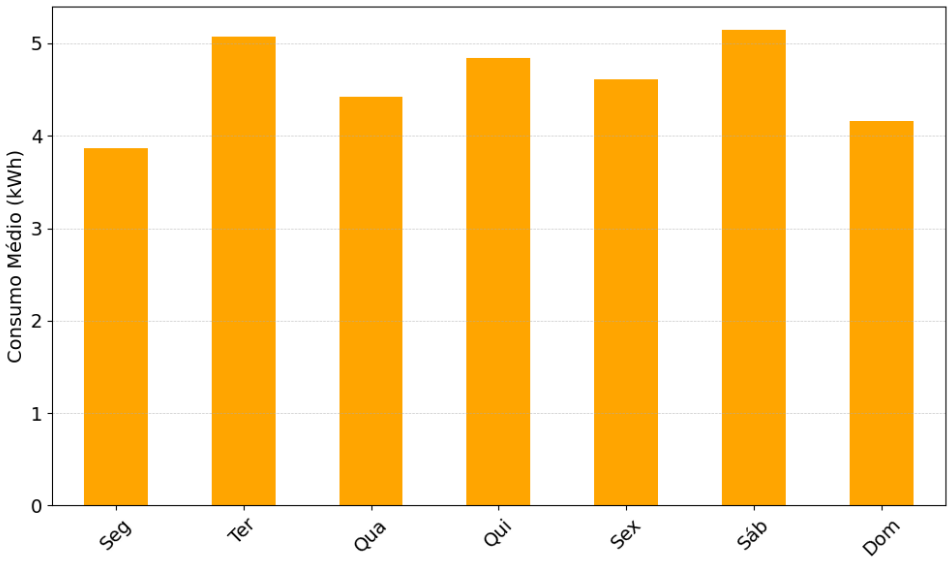


Fonte: Autor(2025)

A linha azul no gráfico representa os erros individuais, enquanto a linha vermelha tracejada indica o Erro Absoluto Médio, que foi de aproximadamente 0,27 kWh. Esse valor sugere que, em média, o modelo erra por 0,27 kWh em suas previsões, o que pode ser considerado relativamente pequeno, dependendo do consumo diário da geladeira. No entanto, observa-se que, em algumas amostras, o erro absoluto ultrapassa 4 kWh, indicando dificuldades do modelo em prever corretamente determinados casos. Esses picos de erro podem estar relacionados a eventos anômalos no consumo, como períodos de alta carga térmica, sobrecargas elétricas repentinas ou outras variações instantâneas na demanda.

Na Figura 31 é ilustrado um gráfico de barras que mostra a média do consumo de energia em kWh para cada dia da semana. No eixo horizontal estão representados os dias da semana, de segunda-feira (Seg) a domingo (Dom), e no eixo vertical está o consumo médio em kWh.

Figura 31 – Média de consumo por dia da semana

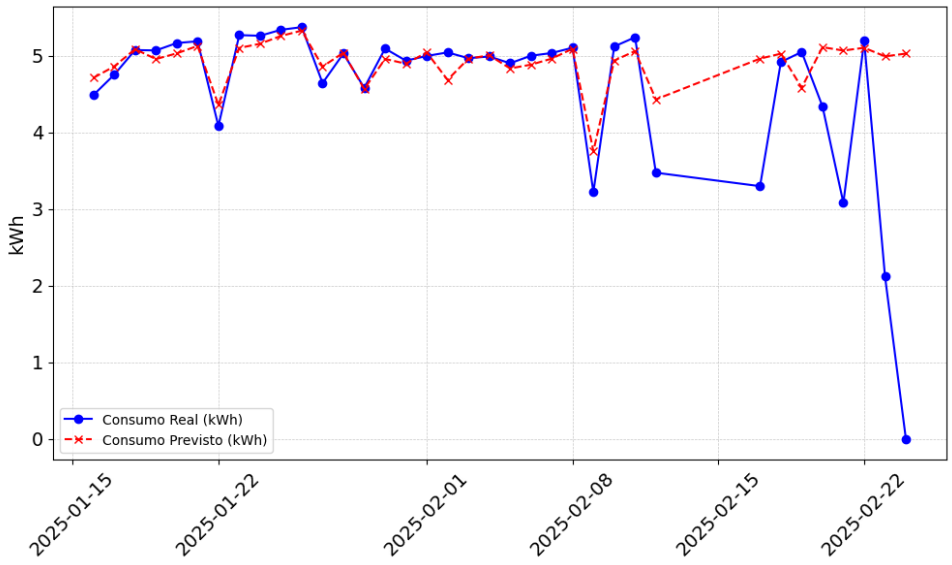


Fonte: Autor(2025)

Este gráfico de barras oferece uma visão clara das flutuações no uso energético do equipamento, evidenciando padrões que podem estar relacionados a hábitos domiciliares. A análise revela que o consumo não é homogêneo ao longo da semana. Notavelmente, terças-feiras e sábados registram os picos de consumo médio, com valores aproximados de 5,1 kWh e 5,15 kWh, respectivamente, o que pode indicar períodos de maior demanda, como o armazenamento de compras ou preparo intensivo de refeições. Por outro lado, quartas-feiras e domingos apresentam os menores índices de consumo, atingindo aproximadamente 4,5 kWh e 4,6 kWh, respectivamente, sugerindo uma menor interação ou carga térmica sobre a geladeira. Essas variações podem ser atribuídas a múltiplos fatores, incluindo a frequência de abertura das portas, a variação da carga térmica dos alimentos e até mesmo as condições ambientais. A identificação desses padrões sazonais no consumo é crucial para o desenvolvimento de estratégias eficazes de gestão energética, permitindo, por exemplo, a implementação de ajustes no termostato em dias de menor demanda ou o planejamento de atividades que influenciam o uso da geladeira para evitar picos desnecessários.

A comparação entre o consumo diário real e o previsto, ilustrada na Figura 32, demonstrou que o modelo XGBoost conseguiu capturar a tendência geral do consumo da geladeira. As curvas de valores reais e previstos apresentaram uma correspondência consistente na maior parte do período analisado. A linha azul contínua representa o consumo

Figura 32 – Consumo Diáro Real *vs* Previsto



Fonte: Autor(2025)

real, ou seja, os dados coletados diretamente da geladeira, enquanto a linha vermelha tracejada indica o consumo previsto pelo modelo, com marcadores em forma de cruz. A análise visual das duas curvas revela que o modelo captura bem a tendência geral do consumo diário, com variações semelhantes ao longo do tempo. No entanto, é possível identificar alguns desvios pontuais, especialmente em momentos de picos ou quedas abruptas no consumo real.

Apesar de observarmos algumas discrepâncias pontuais, especialmente nos dias finais do gráfico onde o consumo real apresentou quedas abruptas não previstas pelo modelo, o Erro Médio Absoluto calculado para todo o período foi de aproximadamente 0.764 kWh. Considerando que a maior parte dos valores de consumo diário no gráfico se situa na faixa de 3 a 5 kWh, um MAE de 0.764 kWh demonstra que, em média, o desvio entre o consumo real e o previsto é relativamente pequeno. Isso valida a percepção de que, para a vasta maioria dos pontos, as duas curvas estão bem alinhadas.

Este MAE, mesmo sendo influenciado pelas grandes diferenças nos dias de anomalia, ainda reflete um bom desempenho médio do modelo *XGBoost* na captura de padrões regulares de consumo. A capacidade do modelo de manter um erro médio abaixo de 1 kWh na maior parte do período analisado, mesmo em um ambiente doméstico com flutuações diárias, evidencia sua eficácia e robustez para a previsão do consumo de energia.

5.1.1 Detecção de anomalias com *Autoencoder* e *One-Class SVM*

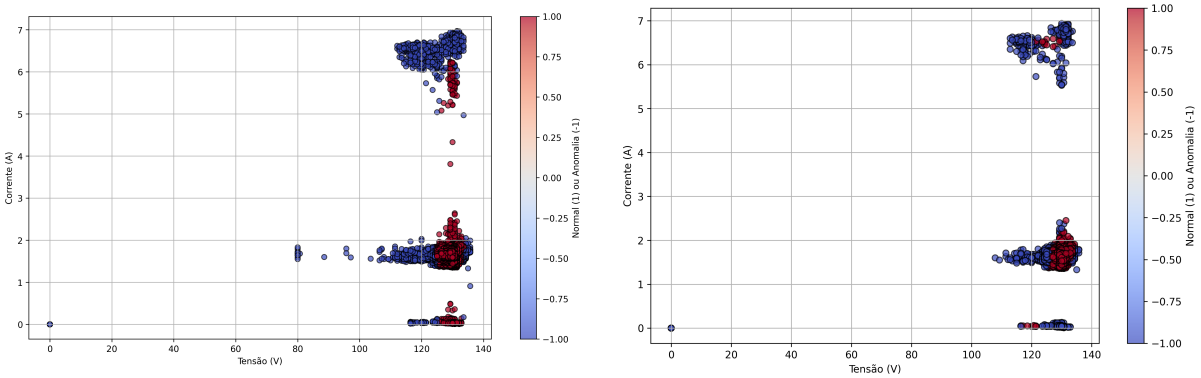
A detecção de anomalias na relação entre tensão (V) e corrente (A) foi realizada utilizando duas técnicas distintas: *One-Class SVM* e *Autoencoder*. Ambas as técnicas

identificam anomalias (azul) e padrões normais (vernelhos), essenciais para monitorar sistemas elétricos e detectar falhas precocemente, como ilustrado na figura 33.

No gráfico, os pontos são coloridos de acordo com o grau de normalidade ou anomalia detectado pelo modelo. Pontos em azul representam comportamentos anomalos, enquanto pontos em vermelho indicam pontos normais. A barra lateral de cores representa essa escala, variando de -1 (anômalo) a 1 (normal).

Os tons mais escuros (tanto azul escuro quanto vermelho escuro) representam regiões onde o modelo está mais confiante na sua classificação — ou seja, são claramente normais ou anômalos. Já os tons mais claros, como azul claro e rosa, refletem regiões de incerteza, próximas à fronteira entre normalidade e anomalia. Esses casos intermediários podem representar estados de transição ou indícios iniciais de falhas, e por isso devem ser analisados com mais atenção.

Essa visualização é especialmente útil para compreender o comportamento da geladeira ao longo do tempo, permitindo identificar momentos em que os padrões de tensão e corrente se desviam do esperado. Assim, a análise gráfica se torna uma ferramenta poderosa tanto para diagnóstico quanto para prevenção de falhas.



(a) One-Class SVM

(b) Autoencoder

Figura 33 – Detecção de anomalias na relação entre tensão (V) e corrente (A) por meio dos modelos One-Class SVM e Autoencoder.

Ao analisar o comportamento dos pontos no gráfico da Figura (a), é possível perceber que a maioria deles está concentrada em regiões específicas, formando agrupamentos densos que representam o padrão habitual de consumo elétrico da geladeira, com corrente entre 1 A e 2,5 A e tensão entre 120 V e 135 V.

Por outro lado, os pontos azuis estão localizados nas extremidades ou de forma isolada, sugerindo que, nesses momentos, houve algum evento incomum. Essas anomalias foram principalmente detectadas em regiões onde a tensão é inferior a 120 V, especialmente quando acompanhada de corrente inferior a 1 A ou superior a 5 A. Tais regiões indicam possíveis falhas momentâneas no funcionamento do motor da geladeira, como picos de corrente, quedas de tensão, ou mesmo interrupções breves no sistema elétrico,

possivelmente relacionadas à instabilidade da rede de alimentação. A abordagem baseada em One-Class SVM tende a agrupar anomalias em regiões bem definidas, o que pode ser vantajoso para identificar eventos elétricos claros e preocupantes.

Já o gráfico da Figura (b), gerado a partir do modelo Autoencoder baseado em redes neurais, funciona de maneira distinta. Ele aprende a representar os dados considerados normais e identifica como anômalos aqueles que não conseguem ser reconstruídos com precisão. Isso permite capturar desvios mais sutis e complexos no comportamento do sistema. As anomalias detectadas pelo Autoencoder também aparecem em regiões de alta tensão (acima de 130 V) e correntes elevadas (superiores a 6 A), semelhantes às observadas pelo *One-Class SVM*. No entanto, também foram encontradas anomalias em regiões de baixa corrente (0 A a 0,5 A) com tensão intermediária, sugerindo consumo atípico ou ineficiente, possivelmente relacionado a estados de repouso anormais ou mau funcionamento de componentes internos.

O *Autoencoder* mostrou-se mais sensível a padrões complexos e menos dependente de limiares rígidos, distribuindo as anomalias de forma mais ampla pelo gráfico. Isso pode ser útil em sistemas onde o comportamento normal não é claramente delimitado, ou onde pequenas variações podem indicar falhas emergentes.

As principais zonas consideradas anômalas estão sumarizadas na Tabela 11, que apresenta as faixas de tensão e corrente associadas a diferentes tipos de anomalias detectadas por cada técnica, bem como possíveis causas ou interpretações técnicas.

Tabela 11 – Zonas consideradas Anomalia

Região	**Faixa de Tensão (V)**	**Faixa de Corrente (A)**	**Técnicas Utilizadas**	**Possível Causa**
Alta Tensão e Alta Corrente	120-140	>5	*One-class* SVM, *Auoenconder*	Picos de consumo, sobrecargas
Tensão com Corrente normal	120-135	1–2,5	*One-class SVM*, *Autoenconder*	Comportamento normal
Corrente e Tensão Anômalo	0–80	0–0.5	*Autoencoder*	Estado de inatividade ou funcionamento anômalo

Fonte: Autor (2025)

Comparando ambas as abordagens, percebe-se que o *one-class SVM* tende a agrupar mais anomalias em regiões específicas, enquanto o Autoencoder distribui as anomalias de forma mais ampla. Isso sugere que o *Autoencoder* pode ser mais eficaz para detectar padrões de anomalias complexos, enquanto o SVM pode ser útil quando há necessidade de uma separação mais bem definida entre normalidade e anomalia. A escolha da melhor técnica depende do objetivo do monitoramento se a intenção for detectar falhas elétricas evidentes, o *one-class SVM* pode ser suficiente por outro lado, se for necessário capturar padrões mais sutis e variações inesperadas, o *Autoencoder* pode oferecer melhores

resultados.

5.1.2 Comparação entre os modelos

Os três modelos foram avaliados em termos de precisão , consumo e de memoria e tempo de inferência no Esp32 como mostrado na Tabela.

Tabela 12 – Comparação de precisão, uso de memória e tempo de inferência no ESP32

Modelo	**Métrica**	**Valor**
XGBoost	MAE	0.27
	Uso de Memória (KB)	128
	Tempo de Inferência (ms)	40
One-Class SVM	Precisão (%)	97%
	Uso de Memória (KB)	90
	Tempo de Inferência (ms)	20
Autoencoder	Precisão (%)	96.5%
	Uso de Memória (KB)	60
	Tempo de Inferência (ms)	10

Fonte: Autor (2025)

O *XGBoost* apresenta boa precisão, mas exige mais memória (128 KB) e possui o maior tempo de inferência (40 ms). É uma ótima opção para previsão de consumo energético, mas pode ser menos eficiente para execução em tempo real devido à latência elevada. O *One-Class SVM* alcançou a maior precisão (97%), com uso moderado de memória e tempo de inferência (20 ms). É adequado para detecção de outliers, oferecendo um bom equilíbrio entre precisão e eficiência computacional. Já o *Autoencoder* apresentou precisão elevada (96,5%), aliada a um menor uso de memória (60 KB) e ao menor tempo de inferência (10 ms). Sua capacidade de modelar padrões complexos o torna ideal para detecção de anomalias em tempo real, desde que o treinamento seja bem ajustado.

A escolha do modelo ideal depende do cenário de aplicação. Para previsão de consumo com alta acurácia em dispositivos com mais recursos, o *XGBoost* se destaca como uma excelente opção. Para detecção de anomalias com equilíbrio entre precisão e custo computacional, o *One-Class SVM* é o mais robusto. Já o *Autoencoder* é altamente indicado para execução em tempo real em ambientes embarcados, graças ao seu baixo uso de memória e rápida inferência — desde que o treinamento seja bem ajustado ao domínio.

5.2 Conclusão

Neste trabalho, foi desenvolvido um protótipo de monitoramento inteligente de energia, empregando o microcontrolador ESP32 e técnicas de TinyML com o objetivo de

realizar previsões de consumo e detectar anomalias em tempo real. A escolha do microcontrolador ESP32 foi motivada por suas características de baixo custo, eficiência energética e capacidade de processamento local, o que elimina a dependência de comunicação contínua com servidores remotos. O sistema utiliza sensores de tensão e corrente para capturar dados precisos sobre o gasto energético, que são processados e analisados diretamente no dispositivo com algoritmos de aprendizado de máquina como XGBoost, One-Class SVM e Autoencoders.

O modelo proposto foi implementado com sucesso em um sistema embarcado de medição inteligente de energia, demonstrando sua adaptação ao TinyML e sua viabilidade para processamento local na borda. Enquanto o XGBoost se destacou como a escolha adequada para a previsão de consumo, apresentando o menor erro médio, embora demande mais recursos computacionais, o Autoencoder, por outro lado, mostrou-se a opção mais eficiente para a detecção de anomalias, com o menor tempo de inferência e consumo de memória, características que o tornam ideal para execução embarcada. Essa combinação de algoritmos permitiu que o sistema monitorasse o consumo de energia de forma eficiente, previsse padrões futuros e identificasse possíveis falhas no gasto energético.

A integração do TinyML no sistema possibilitou análises em tempo real sem a dependência da nuvem, otimizando a eficiência energética e proporcionando insights valiosos para a gestão do consumo de energia. Ademais, a integração com o Node-RED permitiu o desenvolvimento de uma interface de visualização em tempo real, com gráficos e alertas de anomalias, facilitando o gerenciamento energético inteligente. O sistema foi validado através de testes práticos, incluindo a calibração dos sensores e a implementação dos algoritmos de detecção de anomalias, com resultados que confirmaram sua eficácia.

O sucesso desta implementação representa um avanço significativo no uso de *TinyML* para soluções de monitoramento de energia em tempo real, especialmente em ambientes residenciais. Ele demonstra que é possível integrar tecnologias avançadas em dispositivos de baixo custo, resultando em uma solução prática, eficiente e escalável para a otimização do consumo energético

5.3 Sugestões para Trabalhos Futuros

Com base nos resultados obtidos e nas limitações identificadas ao longo deste trabalho, são propostas algumas sugestões para trabalhos futuros que visam aprimorar o protótipo desenvolvido e expandir sua aplicabilidade em diferentes cenários.

A integração do sistema com fontes renováveis de energia, como solar e eólica, pode transformá-lo em uma solução completa de gestão energética, permitindo prever geração e consumo, viabilizar o armazenamento em baterias e aumentar a autonomia do usuário. Além disso, a adaptação para sistemas trifásicos ampliaria sua aplicação em ambientes

comerciais e industriais, exigindo sensores adequados e ajustes nos algoritmos para lidar com medições em três fases, tornando-o mais versátil e robusto.

Nos cenários industriais, o sistema poderia evoluir ainda mais com a incorporação de sensores reforçados e modelos avançados de manutenção preditiva baseados em *machine learning*, permitindo detectar falhas precoces em equipamentos, otimizar processos, reduzir custos operacionais e estender a vida útil das máquinas.

A interface do usuário pode ser melhorada para oferecer maior interatividade, com painéis personalizáveis, notificações automáticas por e-mail ou SMS e integração com assistentes virtuais, como Google Assistant ou Amazon Alexa, facilitando o controle remoto e a tomada de decisão. Quanto aos modelos de aprendizado de máquina, recomenda-se explorar técnicas avançadas de otimização, como *pruning*, quantização e binarização, para reduzir o tamanho dos modelos e adaptá-los melhor a dispositivos embarcados com recursos limitados, como o ESP32. Além disso, redes neurais profundas, como *CNNs* e *RNNs*, podem ser investigadas para aprimorar a detecção de anomalias. Por fim, a implementação de modelos com capacidade de aprendizado contínuo pode permitir que o sistema se adapte automaticamente às mudanças no padrão de consumo ao longo do tempo.

Apêndices

APÊNDICE A – Formulação

```

#include "EmonLib.h"
#include <WiFi.h>
#include <EEPROM.h>
#include <PubSubClient.h>
#include <ArduinoOTA.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "XGBRegressor1.h"
#include "autoencoder_outlier.h"

// TensorFlow Lite Micro
#include "tensorflow/lite/micro/micro_mutable_op_resolver.h"
#include "tensorflow/lite/micro/micro_interpreter.h"
#include "tensorflow/lite/schema/schema_generated.h"
#include "tflite-micro/tensorflow/lite/version.h"

LiquidCrystal_I2C lcd(0x27, 20, 4);
EnergyMonitor SCT013;

char ssid[] = "Digital Connetion_EXT";
char pass[] = "22NKD078002";
const char* mqtt_server = "broker.emqx.io";
WiFiClient espClient;
PubSubClient client(espClient);

int pinSCT = 34;
int tensaoNominal = 127;
double Irms = 0.0;
double potencia = 0.0;

const int pinTensao = 35;
float tensaoRms = 0.0;

#define RELAY_PIN 32

float tarifa = 0.86;
double custo = 0.0;
double kWhTotal = 0.0;

double ultimo_kWh = 0.0;
double ultimo_custo = 0.0;

```

```
44 // TensorFlow Lite Micro
45 constexpr int tensor_arena_size = 16 * 1024;
46 uint8_t tensor_arena[tensor_arena_size];
47 const tflite::Model* model = nullptr;
48 tflite::MicroInterpreter* interpreter = nullptr;
49 TfLiteTensor* input = nullptr;
50 TfLiteTensor* output = nullptr;
51
52 void setupTFLite() {
53   Serial.println("Configurando TensorFlow Lite...");
54   model = tflite::GetModel(autoencoder_outlier);
55   if (model->version() != TFLITE_SCHEMA_VERSION) {
56     Serial.println("Erro: Vers o do modelo TFLite incompat vel!");
57     while (true);
58   }
59
60   static tflite::MicroMutableOpResolver<4> resolver;
61   resolver.AddFullyConnected();
62   resolver.AddReshape();
63   resolver.AddRelu();
64   resolver.AddElu();
65
66   static tflite::MicroInterpreter static_interpreter(model, resolver,
       tensor_arena, tensor_arena_size);
67   interpreter = &static_interpreter;
68
69   TfLiteStatus allocate_status = interpreter->AllocateTensors();
70   if (allocate_status != kTfLiteOk) {
71     Serial.println("Erro: Falha ao alocar tensores!");
72     while (true);
73   }
74
75   input = interpreter->input(0);
76   output = interpreter->output(0);
77
78   if (!input || !output) {
79     Serial.println("Erro: Ponteiros de input/output nulos!");
80     while(true);
81   }
82   Serial.println("TensorFlow Lite configurado com sucesso.");
83 }
84
85 float run_inference(float tensao, float corrente) {
86   input->data.f[0] = (tensao - 120.0f) / 10.0f;
87   input->data.f[1] = corrente / 0.5f;
88
89   TfLiteStatus invoke_status = interpreter->Invoke();
```

```
90    if (invoke_status != kTfLiteOk) {
91      Serial.println("Erro: Falha na infer ncia!");
92      return -1.0f;
93    }
94
95    float diff0 = input->data.f[0] - output->data.f[0];
96    float diff1 = input->data.f[1] - output->data.f[1];
97    return diff0 * diff0 + diff1 * diff1;
98  }
99
100 void connectToWifi() {
101   Serial.println("Conectando ao Wi-Fi...");
102   WiFi.begin(ssid, pass);
103   int attempts = 0;
104   while (WiFi.status() != WL_CONNECTED && attempts < 20) {
105     delay(500);
106     Serial.print(".");
107     attempts++;
108   }
109   if (WiFi.status() == WL_CONNECTED) {
110     Serial.println("\nWi-Fi conectado.");
111     Serial.print("Endere o IP: ");
112     Serial.println(WiFi.localIP());
113   } else {
114     Serial.println("\nFalha ao conectar ao Wi-Fi. Reiniciando...");
115     delay(5000);
116     ESP.restart();
117   }
118 }
119
120 void callback(char* topic, byte* payload, unsigned int length) {
121   String message;
122   for (int i = 0; i < length; i++) {
123     message += (char)payload[i];
124   }
125
126   if (String(topic) == "cmd/rele") {
127     if (message == "ON") {
128       digitalWrite(RELAY_PIN, HIGH);
129     } else if (message == "OFF") {
130       digitalWrite(RELAY_PIN, LOW);
131     }
132   }
133 }
134
135 void reconnect() {
136   while (!client.connected()) {
```

```
137     if (client.connect("Leitor_Corrente_ESP32")) {
138       client.subscribe("cmd/rele");
139     } else {
140       delay(5000);
141     }
142   }
143 }
144 
145 void setup() {
146   Serial.begin(115200);
147   connectToWifi();
148   client.setServer(mqtt_server, 1883);
149   client.setCallback(callback);
150 
151   SCT013.current(pinSCT, 1.220);
152   pinMode(RELAY_PIN, OUTPUT);
153   digitalWrite(RELAY_PIN, LOW);
154 
155   lcd.begin();
156   lcd.backlight();
157   lcd.clear();
158   lcd.print("Iniciando...");
159 
160   ArduinoOTA.begin();
161 
162   EEPROM.begin(16);
163   EEPROM.get(0, kWhTotal);
164   EEPROM.get(8, custo);
165   ultimo_kWh = kWhTotal;
166   ultimo_custo = custo;
167 
168   setupTFLite();
169 }
170 
171 void loop() {
172   if (!client.connected()) reconnect();
173   client.loop();
174   ArduinoOTA.handle();
175 
176   Irms = SCT013.calcIrms(1480);
177   potencia = Irms * tensaoNominal;
178 
179   int menorValor = 4095;
180   for (int i = 0; i < 3000; i++) {
181     int valor_inst = analogRead(pinTensao);
182     if (valor_inst < menorValor) menorValor = valor_inst;
183     delayMicroseconds(1);
```

```
184   }
185
186   float tensaoPico = map(menorValor, 0, 4095, 112, 220);
187   tensaoRms = tensaoPico / 1.414;
188
189   double potenciaKW = potencia / 1000.0;
190   kWhTotal += potenciaKW * (1.0 / 3600.0);
191   custo += potenciaKW * tarifa * (1.0 / 3600.0);
192
193   if (abs(kWhTotal - ultimo_kWh) > 0.001 || abs(custo - ultimo_custo) >
          0.01) {
194     EEPROM.put(0, kWhTotal);
195     EEPROM.put(8, custo);
196     EEPROM.commit();
197     ultimo_kWh = kWhTotal;
198     ultimo_custo = custo;
199   }
200
201   double input_xgb[4] = {tensaoRms, Irms, potencia, millis() / 1000.0};
202   double resultado = score(input_xgb);
203
204   float mse = run_inference(tensaoRms, Irms);
205   const float MSE_THRESHOLD = 3.0;
206   const float MIN_CURRENT_THRESHOLD = 1.0;
207   const float MAX_CURRENT_THRESHOLD = 2.2;
208
209   bool is_anomalia_ae = mse > MSE_THRESHOLD || Irms <
          MIN_CURRENT_THRESHOLD || Irms > MAX_CURRENT_THRESHOLD;
210
211   char msg_tensao[30], msg_corrente[30], msg_potencia[30],
          msg_custo[30], msg_kWhTotal[30];
212   sprintf(msg_tensao, "%.2f", tensaoRms);
213   sprintf(msg_corrente, "%.2f", Irms);
214   sprintf(msg_potencia, "%.2f", potencia);
215   sprintf(msg_custo, "%.6f", custo);
216   sprintf(msg_kWhTotal, "%.6f", kWhTotal);
217
218   client.publish("sensor/tensao", msg_tensao);
219   client.publish("sensor/corrente", msg_corrente);
220   client.publish("sensor/potencia", msg_potencia);
221   client.publish("sensor/custo", msg_custo);
222   client.publish("sensor/kWhTotal", msg_kWhTotal);
223   client.publish("sensor/ml_inferencia", String(resultado, 4).c_str());
224   client.publish("sensor/ae_mse", String(mse, 4).c_str());
225   client.publish("sensor/ae_anomalia", is_anomalia_ae ? "1" : "0");
226
227   lcd.clear();
```

```
228   lcd.setCursor(0, 0); lcd.print("XGB: "); lcd.print(resultado, 4);
229   lcd.setCursor(0, 1); lcd.print("AE: "); lcd.print(is_anomalia_ae ?
          "ANOMALIA" : "NORMAL");
230   lcd.setCursor(0, 2); lcd.print("T:"); lcd.print(tensaoRms, 2);
          lcd.print("V I:"); lcd.print(Irms, 2); lcd.print("A");
231   lcd.setCursor(0, 3); lcd.print("kWh:"); lcd.print(kWhTotal, 3);
          lcd.print(" R$");
232
233   Serial.printf("Irms: %.3f A | Tens o: %.2f V | Pot ncia: %.4f kW |
          kWh: %.6f\n", Irms, tensaoRms, potenciaKW, kWhTotal);
234   delay(1000);
235 }
```

Listing A.1 – Código para monitoramento de Consumo de energia com ESP32.

APÊNDICE B – Custo do Protótipo

Este apêndice apresenta o detalhamento dos custos estimados para a montagem do protótipo desenvolvido nesta dissertação. Os valores referem-se aos preços de mercado obtidos em maio de 2025 e incluem todos os componentes de hardware utilizados no projeto, permitindo demonstrar a viabilidade econômica do sistema proposto.

Tabela 13 – Custos estimados do protótipo desenvolvido

Item	**Qtd**	**Unitário (R$)**	**Total (R$)**
Kit jumpers (30 cabos)	1 kit	33,41	33,41
Resistor 10k (CR25, 1/4W, 100 peças)	1 kit	12,02	12,02
SCT-013 (30A/1V)	1 unid.	62,90	62,90
ESP32	1 unid.	39,99	39,99
Sensor ZMPT101B	1 unid.	36,22	36,22
Fonte ajustável (Protoboard 3.3V/5V)	1 unid.	15,80	15,80
Módulo relé de 2 canais	1 unid.	14,99	14,99
Protoboard	2 unid.	17,39	34,78
Carregador tomada (fonte USB 5V 2.1A)	1 unid.	19,90	19,90
Cabo USB macho 3.0 (base cooler)	1 unid.	30,90	30,90
Capacitor (kit com 25 peças)	1 kit	9,97	9,97
Display LCD 20x4	1 unid.	63,90	63,90
Kit resistor (300 peças)	1 kit	37,32	37,32
Alicate amperímetro Exbom MD-Y400	1 unid.	28,00	28,00
Total geral			**459,10**

Fonte: Autor(2025)

Anexos

ANEXO A – Artigos publicados

Durante o desenvolvimento desta dissertação, foram publicados artigos científicos que fortalecem os fundamentos teóricos e práticos abordados no estudo.

- NAMBUNDO, J. M.; GOMES, O. de S. M.; SOUZA, A. D. de; MACHADO, R. C. S. Cybersecurity and major cyber threats of smart meters: a systematic mapping review. Energies, Basel, v. 18, n. 6, art. 1445, 2025. DOI: 10.3390/en18061445. Disponível em: https://doi.org/10.3390/en18061445.
- NAMBUNDO, J. M.; GOMES, O. de S. M. Sistema de Análise Preditiva em Tempo Real para Smart Meters usando Machine Learning Embarcado (TinyML). Anais do XI Simpósio Brasileiro de Sistemas Elétricos (SBSE 2025), São João del-Rei – MG, 29 de julho a 1º de agosto de 2025. Artigo aceito para apresentação e publicação.

Referências

1 Ministério de Minas e Energia (MME); Empresa de Pesquisa Energética (EPE); CNN Brasil. Participação de fontes renováveis na matriz elétrica brasileira. *Relatórios e Dados Oficiais*, 2024. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-819/topico-715/BEN_S%C3%ADntese_2024_PT.pdf>. 15, 16

2 ŽIVIC, N. S.; UR-REHMAN, O.; RULAND, C. Evolution of smart metering systems. In: *2015 23rd Telecommunications Forum Telfor (TELFOR)*. [s.n.], 2015. p. 635–638. Disponível em: <https://doi.org/10.1109/TELFOR.2015.7377547>. 15

3 ALVES, F. C. L.; JUNIOR, A. O. P.; SáNCHEZ, J. C. M. Analysis of regulatory process for the implementation of smart metering in brazil. *Decision Analytics Journal*, v. 3, p. 100053, 2022. Disponível em: <https://doi.org/10.1016/j.dajour.2022.100053>. 15

4 SIRYANI, J.; TANJU, B.; EVELEIGH, T. J. A machine learning decision-support system improves the internet of things' smart meter operations. *IEEE Internet of Things Journal*, v. 4, n. 4, p. 1056–1066, 2017. 15

5 HAYASHI, V. T. et al. B2b b2c architecture for smart meters using iot and machine learning: a brazilian case study. In: *2020 International Conference on Smart Grids and Energy Systems (SGES)*. [s.n.], 2020. p. 826–831. Acesso em: 23 out. 2024. Disponível em: <https://api.semanticscholar.org/CorpusID:232149602>. 15

6 KALLIMANI, R. et al. Tinyml: Tools, applications, challenges, and future research directions. *ArXiv*, abs/2303.13569, 2023. Acesso em: 23 out. 2024. Disponível em: <https://api.semanticscholar.org/CorpusID:257756932>. 15, 16, 17

7 GARCIA, M. Cybersecurity in smart grids: Challenges and solutions. *IEEE Security & Privacy*, v. 5, n. 1, p. 24–37, 2023. 16

8 ABRAHAM, G. Identificação de fraudes de energia elétrica em consumidores comerciais - uma aplicação voltada aos medidores inteligentes. In: *Anais Do Computer on the Beach*. [S.l.: s.n.], 2018. 16

9 LI, X.; WANG, Y. Integration of federated machine learning in smart metering systems. *IEEE Transactions on Industrial Informatics*, v. 19, n. 5, p. 4328–4337, 2024. Disponível em: <https://ieeexplore.ieee.org/document/9942380>. 16

10 MATOS, Y. C. C. Detecção de fraudes no consumo de energia elétrica usando árvore de decisão. Belém, 2017. 16

11 WANG, H.; ZHANG, L. Smart electricity meter data intelligence for future energy systems: A comprehensive review. *IEEE Transactions on Industrial Informatics*, v. 9, n. 6, p. 710–721, 2023. Disponível em: <https://ieeexplore.ieee.org/document/7063262>. 16

12 KIM, J.; LEE, S. Deep learning based smart meter data analytics for electricity load forecasting. *IEEE Access*, v. 10, p. 3301–3312, 2023. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9167704>. 16

13 MARTINEZ, D.; JOHNSON, T. Analyzing consumer behavior insights from smart meter data: A comprehensive review. *IEEE Transactions on Smart Grid*, v. 14, n. 3, p. 1742–1755, 2023. Disponível em: <https://ieeexplore.ieee.org/document/10607792>. 19

14 WANG, Y. et al. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, v. 10, n. 3, p. 3125–3148, 2019. 19, 24

15 International Energy Agency. *Energy Efficiency 2023*. Paris, 2023. Accessed: 2025-05-17. Disponível em: <https://www.iea.org/reports/energy-efficiency-2023>. 19, 20, 24

16 WORRELL, E. et al. Industrial energy efficiency and climate change mitigation. *Energy Efficiency*, v. 1, n. 1, p. 87–97, 2009. Disponível em: <https://www.researchgate.net/publication/225336407>. 20

17 National Institute of Standards and Technology (NIST). *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 4.0*. 2014. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1108r4.pdf>. Accessed: 2025-07-22. 20

18 CECATI, C. et al. An overview on the smart grid concept. *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, p. 3322–3327, 2010. 20

19 Operador Nacional do Sistema Elétrico (ONS). *Plano da Operação Elétrica de Médio Prazo do SIN – PAR/PEL 2023, Ciclo 2024-2028*. São Paulo, 2023. Acesso em: 17 maio 2025. Disponível em: <https://www.ons.org.br>. 21

20 AMIN, S. M.; WOLLENBERG, B. Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine*, v. 3, n. 5, p. 34–41, 2005. 21

21 GUNGOR, V. C.; LU, B.; HANCKE, G. P. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Transactions on Industrial Electronics*, v. 57, n. 10, p. 3557–3564, 2010. 21

22 RODRíGUEZ-MOLINA, J. et al. Business models in the smart grid: Challenges, opportunities and proposals for prosumer profitability. *Energies*, v. 7, n. 9, p. 6142–6171, 2014. 21

23 KUMAR, P. et al. Smart grid metering networks: A survey on security, privacy and open research issues. *IEEE Communications Surveys & Tutorials*, v. 21, n. 3, p. 2886–2927, 2019. 21

24 NEVES, L. C.; BAGAROLLI, A. Os desafios da implementação dos projetos-piloto de smart grid no brasil. *Cadernos CPqD de Tecnologia*, Campinas, v. 9, n. 1, p. 15–22, 2013. 22

25 SILVA, J. Advanced metering infrastructure in smart grids: A comprehensive review. *IEEE Transactions on Smart Grid*, v. 12, n. 4, p. 310–322, 2022. 22

26 CARVALHO, M.; LIMA, R. Energy management through advanced metering infrastructure. *IEEE Power and Energy Magazine*, v. 19, n. 3, p. 45–53, 2021. 22

27 RODRIGUES L., S. P.; ALMEIDA, T. Real-time monitoring in ami systems: Challenges and opportunities. *IEEE Access*, v. 11, p. 8905–8916, 2023. 22

28 MARTINS, F. Dynamic pricing in smart grids supported by ami technologies. *IEEE Transactions on Power Systems*, v. 39, n. 2, p. 1101–1110, 2024. 23

29 PEREIRA, A.; COSTA, E. Consumer behavior and energy savings in smart grids with ami. *IEEE Consumer Electronics Magazine*, v. 12, n. 5, p. 34–42, 2023. 23

30 SANTOS, G.; NOGUEIRA, M. Overcoming communication challenges in advanced metering infrastructure. *IEEE Communications Surveys & Tutorials*, v. 26, n. 1, p. 120–137, 2024. 23

31 CECATI, C. et al. An overview on the smart grid concept. *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, p. 3322–3327, 2010. 23, 24

32 COOK, B. et al. The smart meter and a smarter consumer: quantifying the benefits of smart meter implementation in the united states. *Chemistry Central Journal*, Springer, v. 6, p. 1–16, 2012. 23

33 PIRBAZARI, A. M. et al. Short-term load forecasting using smart meter data: A generalization analysis. *Processes*, v. 8, n. 4, p. 484, 2020. 24

34 COSTA, G.; (COORD.), M. I. *Machine Learning e Big Data no Setor Elétrico: Guia de Conteúdo para Docentes*. Brasil, 2023. Coordenação técnica: Caroline Luciane Broering Dutra e Roberta Hessmann Knopki. Disponível em: <https://www.giz.de/de/downloads/giz2023-machine-learning-setor-eletrico.pdf>. 24, 27

35 UK Government Publishing Service. Smart meter statistics in great britain: Quarterly report to end december 2023. *Government Publishing Service*, Government Publishing Service, p. 1–, 2023. Disponível em: <https://assets.publishing.service.gov.uk/media/65fc3d0a65ca2f001b7da7c5/Q4_2023_Smart_Meters_Statistics_Report.pdf>. Acesso em: 4 nov. 2024. 25

36 RAZA, M. H. et al. *Smart Meters for Smart Energy: A Review of Business Intelligence Applications*. 2023. 120001-120022 p. 25

37 smart-meter-adoption. Iot analytics (2023). smart meter adoption. *IEA Report*, 2023. Disponível em: <https://iot-analytics.com/>. 25

38 ASGHAR, M. R.; LJUNGGREN, P.; KARLSSON, P. Security and privacy in smart metering systems: A systematic mapping study. *Journal of Cleaner Production*, Elsevier, v. 137, p. 1121–1136, 2017. 25

39 WANG, W.; XU, Y.; KHANNA, M. A survey on the communication architectures in smart grid. *Computer Networks*, Elsevier, v. 55, n. 15, p. 3604–3629, 2018. 25

40 CHAUDHARI, B.; ZENNARO, M.; BORKAR, S. Lpwan technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet*, MDPI, v. 12, n. 3, p. 46, 2020. 25

41 SANKAR, L. et al. Smart meter privacy: A theoretical framework. *IEEE Transactions on Smart Grid*, IEEE, v. 4, n. 2, p. 837–846, 2012. 26

42 YAN, Y. et al. A survey on smart grid communication infrastructures: Motivations, requirements and challenges. *IEEE Communications Surveys Tutorials*, v. 15, n. 1, p. 5–20, 2013. 26

43 COSTA, G.; (COORD.), M. I. *Machine Learning e Big Data no Setor Elétrico: Guia de Conteúdo para Docentes*. Brasil, 2023. Coordenação técnica: Caroline Luciane Broering Dutra e Roberta Hessmann Knopki. Disponível em: <https://www.giz.de/de/downloads/giz2023-machine-learning-setor-eletrico.pdf>. 26

44 SINGH, R.; VERMA, A. Machine learning and big data analytics for smart grids: Challenges and opportunities. *IEEE Transactions on Smart Grid*, v. 13, n. 6, p. 4501–4512, 2022. 26, 28

45 MOHANTY, A. et al. Smart grid and application of big data: Opportunities and challenges. *Sustainable Energy Technologies and Assessments*, v. 71, p. 104011, 2024. 26

46 SINGH, T.; S, D. V. A modern data architecture with apache hadoop. p. 574–579, 2015. 27

47 WANG, K.; KHAN, M. M. H. Performance prediction for apache spark platform. In: *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*. [S.l.: s.n.], 2015. p. 166–173. 28, 29

48 SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, v. 2, p. 160, 2021. 28, 29

49 DOE, J.; SMITH, A.; JOHNSON, M. Anomaly detection with machine learning algorithms and big data in electricity consumption. *Journal of Big Data Analytics in Energy*, v. 12, n. 4, p. 567–578, 2023. 28, 29

50 MNIH, V.; KAVUKCUOGLU, K.; SILVER, D. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 29, 30

51 KUMAR, A.; JAIN, S.; SINGH, R. A survey on machine learning approaches and techniques. In: *2020 IEEE Students Conference on Engineering & Systems (SCES)*. [S.l.: s.n.], 2020. p. 1–6. 30

52 PECIOSKI, D. et al. An overview of reinforcement learning techniques. In: *2023 12th Mediterranean Conference on Embedded Computing (MECO)*. [S.l.: s.n.], 2023. p. 1–4. 30

53 SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. 2nd. ed. MIT Press, 2018. Disponível em: <http://incompleteideas.net/book/the-book-2nd.html>. 30

54 WATKINS, C. J.; DAYAN, P. Q-learning. *Machine Learning*, Springer, v. 8, n. 3-4, p. 279–292, 1992. 30

55 MNIH, V. et al. Human-level control through deep reinforcement learning. *Nature*, Nature Publishing Group, v. 518, n. 7540, p. 529–533, 2015. 30

56 KUMAR, D.; MALLICK, P.; TRIPATHY, S. Machine learning and deep learning applications in smart grid: A comprehensive review. *Computer Science Review*, Elsevier, v. 43, p. 100456, 2022. 30

57 ARULKUMARAN, K. et al. A brief survey of deep reinforcement learning. *IEEE Signal Processing Magazine*, v. 34, n. 6, p. 26–38, 2017. 30

58 YIP, S.-C. et al. An anomaly detection framework for identifying energy theft and defective meters in smart grids. *International Journal of Electrical Power & Energy Systems*, v. 101, p. 189–203, 2018. 31

59 HAYASHI, V. et al. B2b b2c architecture for smart meters using iot and machine learning: a brazilian case study. In: *2020 International Conference on Smart Grids and Energy Systems (SGES)*. [S.l.: s.n.], 2020. p. 826–831. 31

60 AGHERA, R. et al. *A Deep Learning Technique using Low Sampling rate for residential Non Intrusive Load Monitoring*. 2021. <https://arxiv.org/abs/2111.05120>. ArXiv preprint arXiv:2111.05120. Disponível em: <https://arxiv.org/abs/2111.05120>. 31

61 ANNADATA, Y. S. H.; MOGANARENGAM, V.; NIKOUBIN, T. Tinyml powered drone in agriculture application. *2024 IEEE 17th Dallas Circuits and Systems Conference (DCAS)*, p. 1–6, 2024. 31

62 SURESH, V. M. et al. Powering iot through embedded machine learning and lora. *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, p. 349–354, 2018. 31

63 MOALLEMI, A. et al. Exploring scalable, distributed real-time anomaly detection for bridge health monitoring. *IEEE Internet of Things Journal*, v. 9, p. 17660–17674, 2022. 31, 32

64 WARDANA, I. N. K.; FAHMY, S. A.; GARDNER, J. W. Tinyml models for a low-cost air quality monitoring device. *IEEE Sensors Letters*, v. 7, p. 1–4, 2023. 31, 32

65 HAYAJNEH, A. M. et al. Tiny machine learning on the edge: A framework for transfer learning empowered unmanned aerial vehicle assisted smart farming. *IET Smart Cities*, v. 6, p. 10–26, 2024. 31, 32

66 RASHID, R. A. et al. Machine learning for smart energy monitoring of home appliances using iot. *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, p. 66–71, 2019. 31, 32

67 NAMBUNDO, J. et al. Cybersecurity and major cyber threats of smart meters: A systematic mapping review. *Energies*, v. 18, n. 6, 2025. Disponível em: <https://doi.org/10.3390/en18061445>. 32

68 SAVADATTI, M. B. et al. An overview of predictive analysis based on machine learning techniques. In: *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. [S.l.: s.n.], 2022. p. 1–6. 33

69 BOKONDA, P. L.; OUAZZANI-TOUHAMI, K.; SOUISSI, N. Predictive analysis using machine learning: Review of trends and methods. In: *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*. [S.l.: s.n.], 2020. p. 1–6. 33

70 SAMSON-ONUORAH, C. I.; BAKARE, F. A. Dynamic strategic foresight using predictive business analytics: Modeling competitive advantage under volatile market and technological conditions. *International Journal of Research Publication and Reviews*, International Journal of Research Publication and Reviews, v. 6, n. 5, p. 1044–1060, 2025. Disponível em: <https://doi.org/10.55248/gengpi.6.0525.1627>. 33

71 VAPNIK, V.; BOSER, B. A training algorithm for optimal margin classifiers. *Journal of Machine Learning*, p. 144–152, 1992. 34

72 Data Science Academy. *Capítulo 6 – O perceptron – parte 1*. 2007. <https://www.deeplearningbook.com.br/o-perceptron-parte-2/>. Acesso em: 21 maio 2025. 35

73 HAYKIN, S. *Neural Networks and Learning Machines*. 3rd. ed. [S.l.]: Prentice Hall, 1999. 34

74 VIDHYA, A. *One-Class SVM for Anomaly Detection*. 2024. Disponível em: <https://www.analyticsvidhya.com/blog/2024/03/one-class-svm-for-anomaly-detection/>. Acesso em: março de 2024. 35

75 LI, K.-L. et al. Improving one-class svm for anomaly detection. In: *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*. [S.l.: s.n.], 2003. v. 5, p. 3077–3081 Vol.5. 35

76 KHAN, N. et al. Efficient anomaly detection for high-dimensional sensing data with one-class support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 34, n. 11, p. 5216–5229, 2022. 36

77 CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.]: ACM, 2016. p. 785–794. 36

78 CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. 36

79 ANJOS, G. V. Lima dos. *Sistema Inteligente para Identificação de Suspeitos de Fraude no Consumo de Água*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro – PUC, 2022. 36

80 ARAUJO, B. S. d.; ALMEIDA, H. L. S. d. Métodos de inteligência computacional aplicados à detecção de fraude de consumidores de energia elétrica. In: *IEEE Latin America*. [S.l.: s.n.], 2019. 36

81 KRAMER, M. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, v. 37, n. 12, p. 233–243, 1991. 37

82 HINTON, G.; SALAKHUTDINOV, R. Reducing the dimensionality of data with neural networks. *Science*, v. 313, p. 504–507, 2006. 37

83 PEREIRA, J.; SILVEIRA, M. Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In: *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. [s.n.], 2018. p. 1275–1282. Disponível em: <https://doi.org/10.1109/ICMLA.2018.0020>. 37

84 GeeksforGeeks Contributors. *Types of Autoencoders*. 2025. Acessado em: 21 de maio de 2025. Disponível em: <https://www.geeksforgeeks.org/types-of-autoencoders/>. 38

85 ZHAI, S. et al. Deep structured energy based models for anomaly detection. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, p. 1100–1109, 2016. 38

86 ZHAI, S. et al. Deep structured energy based models for anomaly detection. In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. New York, NY, USA: JMLR: W&CP volume 48, 2016. Disponível em: <https://arxiv.org/abs/1605.07717>. 39

87 SAKURADA, M.; YAIRI, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: ACM. *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. [S.l.], 2014. p. 4–11. 39

88 ELHANASHI, A. et al. Advancements in tinyml: Applications, limitations, and impact on iot devices. *Electronics*, v. 13, n. 17, 2024. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/13/17/3562>. 39, 45

89 ABADADE, Y. et al. A comprehensive survey on tinyml. *IEEE Access*, v. 11, p. 96892–96922, 2023. 39

90 JOUINI, O. et al. A survey of machine learning in edge computing: Techniques, frameworks, applications, issues, and research directions. *Technologies*, v. 12, n. 6, 2024. ISSN 2227-7080. Disponível em: <https://www.mdpi.com/2227-7080/12/6/81>. 39

91 MIHIGO, I. N. et al. On-device iot-based predictive maintenance analytics model: Comparing tinylstm and tinymodel from edge impulse. *Sensors*, MDPI, v. 22, n. 14, p. 5174, 2022. 40

92 KALLIMANI, R. et al. Tinyml: Tools, applications, challenges, and future research directions. *Multimedia Tools and Applications*, v. 83, p. 29015–29045, 2024. Disponível em: <https://doi.org/10.1007/s11042-023-16740-9>. 40

93 HUANG, X. et al. High drive and low leakage current mbc fet with channel thickness 1.2nm/0.6nm. In: *2020 IEEE International Electron Devices Meeting (IEDM)*. [S.l.: s.n.], 2020. p. 12.1.1–12.1.4. 40

94 MATOS, M. et al. A machine learning based energy management system for renewable energy communities. In: *2023 IEEE 3rd International Conference on Industrial Electronics for Sustainable Energy Systems (IESES)*. [S.l.: s.n.], 2023. p. 1–6. 40

95 CHARAN, K. S. An auto-encoder based tinyml approach for real-time anomaly detection. *SAE International Journal of Advances and Current Practices in Mobility*, v. 5, n. 2022-28-0406, p. 1496–1501, 2022. 40

96 DIAB, M. S.; RODRIGUEZ-VILLEGAS, E. Embedded machine learning using microcontrollers in wearable and ambulatory systems for health and care applications: A review. *IEEE Access*, v. 10, p. 98450–98474, 2022. 41

97 ZHANG, S. et al. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors*, MDPI, v. 22, n. 4, p. 1476, 2022. 41

98 PAUL, A. J.; MOHAN, P.; SEHGAL, S. Rethinking generalization in american sign language prediction for edge devices with extremely low memory footprint. In: *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. [S.l.: s.n.], 2020. p. 147–152. 41

99 MOHAN, P.; PAUL, A. J.; CHIRANIA, A. A tiny cnn architecture for medical face mask detection for resource-constrained endpoints. In: *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2021*. [S.l.]: Springer, 2021. p. 657–670. 41

100 JACOB, B. et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *arXiv:1712.05877*, 2018. 41

101 HAN, S. et al. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *ICLR 2016*, 2015. 41

102 ZMORA, H. W. N.; RODGE, J. *Achieving FP32 Accuracy for INT8 Inference Using Quantization Aware Training with NVIDIA TensorRT*. <https://developer.nvidia.com/blog/achieving-fp32-accuracy-for-int8-inference-using-quantization-aware-training-with-tensorrt/>. Acesso em: 22 maio 2025. 41, 42

103 HAN, S.; MAO, H.; DALLY, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: *International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2016. 42

104 HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 42

105 HAN, S.; MAO, H.; DALLY, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. [s.n.], 2015. Disponível em: <https://arxiv.org/abs/1510.00149>. 42

106 ZIV, Y.; GOLDBERGER, J.; RAVIV, T. R. Stochastic weight pruning and the role of regularization in shaping network structure. *Neurocomputing*, Elsevier, v. 456, p. 1–12, 2021. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0925231221011917>. 43

107 CHIAO, R. A. A. *TensorFlow Model Optimization Toolkit — Weight Clustering API*. <https://blog.tensorflow.org/2020/08/tensorflow-model-optimization-toolkit-weight-clustering-api.html>. Acesso em: 22 maio 2025. 43, 44

108 HAN, S.; MAO, H.; DALLY, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In: *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. San Juan, Puerto Rico: [s.n.], 2016. Disponível em: <https://arxiv.org/abs/1510.00149>. 44

109 SCHIZAS, N. et al. Tinyml for ultra-low power ai and large scale iot deployments: A systematic review. *Future Internet*, v. 14, n. 12, 2022. ISSN 1999-5903. Disponível em: <https://www.mdpi.com/1999-5903/14/12/363>. 45

110 LIN, J. et al. Tiny machine learning: Progress and futures. 2024. ArXiv:2403.19076v2, 29 Mar 2024. Disponível em: <https://tinyml.mit.edu>. 45

111 RAHMAN, M. A. et al. Scalable machine learning-based intrusion detection system for iot-enabled smart cities. *Sustainable Cities and Society*, Elsevier, v. 62, p. 102324, 2020. Disponível em: <https://doi.org/10.1016/j.scs.2020.102324>. 45

112 HART, D. W. *Eletrônica de potência: análise e projetos de circuitos*. [S.l.]: McGraw Hill Brasil, 2016. P. 9–10. 46, 47

113 SILVA, I. G. d. et al. Uma abordagem estatística para a previsão de potência reativa em sistemas elétricos. *Pesquisa Operacional*, FAPESP, v. 26, n. 2, p. 289–301, ago 2006. 47

114 SADIKU, M. N. O.; ALEXANDER, C. *Fundamentos de Circuitos Elétricos*. Porto Alegre: Bookman, 2011. 52, 63

115 ENGINEERS, L. M. *ESP32 Pinout Reference – GPIO, ADC, DAC, Touch, SPI, I2C, UART, PWM, Power, GND, EN Pins*. 2025. Acesso em: 22 maio 2025. Disponível em: <https://lastminuteengineers.com/esp32-pinout-reference/>. 52

116 SYSTEMS, E. *ESP32 Series Datasheet*. 2025. <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 22 maio 2025. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. 53

117 MóDULO Sensor de Tensão AC 0 a 250V - Voltímetro ZMPT101B. [S.l.]: Eletrogate, 2025. <https://www.eletrogate.com/modulo-sensor-de-tensao-ac-0-a-250v-voltimetro-zmpt101b>. Acesso em: 22 de maio de 2025. 53, 54, 62

118 Vida de Silício. *SCT-013 - Sensor de Corrente Alternada*. 2025. <https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/>. Acesso em: 22 de maio de 2025. 54

119 CHOI, I.; SHIM, H.; CHANG, N. Low-power color tft lcd display for hand-held embedded systems. In: *Proceedings of the 2002 International Symposium on Low Power Electronics and Design*. New York, NY, USA: Association for Computing Machinery, 2002. (ISLPED '02), p. 112–117. ISBN 1581134754. Disponível em: <https://doi.org/10.1145/566408.566440>. 55

120 MakerHero. *Display LCD 20x4 Backlight Azul*. 2025. <https://www.makerhero.com/produto/display-lcd-20x4-backlight-azul/>. Acesso em: 22 de maio de 2025. 55

121 JABBAR, W. A. et al. Design and implementation of iot-based automation system for smart home. In: *2018 International Symposium on Networks, Computers and Communications (ISNCC)*. [S.l.: s.n.], 2018. p. 1–6. 55

122 Eletrogate. *Módulo Relé 2 Canais 5V*. 2025. <https://www.eletrogate.com/modulo-rele-2-canais-5v>. Acesso em: 22 de maio de 2025. 56

123 FONTE Ajustável para Protoboard. [S.l.]: Eletrogate, 2025. <https://www.eletrogate.com/fonte-ajustavel-para-protoboard>. Acesso em: 22 de maio de 2025. 57

124 SEDRA, A. S.; SMITH, K. C. *Microelectronic Circuits*. 7th. ed. New York: Oxford University Press, 2015. 57

125 MakerHero. *Resistor - Guia de Componentes Eletrônicos*. 2025. <https://www.makerhero.com/guia/componentes-eletronicos/resistor/>. Acesso em: 22 de maio de 2025. 58

126 MakerHero. *Capacitor - Guia de Componentes Eletrônicos*. 2025. <https://www.makerhero.com/guia/componentes-eletronicos/capacitor/>. Acesso em: 22 de maio de 2025. 58

127 MakerHero. *Como funciona uma Protoboard?* 2025. <https://www.makerhero.com/blog/como-funciona-uma-protoboard/>. Acesso em: 22 de maio de 2025. 58

128 Arduino. *Arduino Reference - map()*. 2023. <https://www.arduino.cc/reference/en/language/functions/math/map/>. Acesso em: 23 maio 2025. 63

129 Vida de Silício. *SCT-013 Sensor de Corrente Alternada*. 2025. <https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/>. Acesso em: 23 maio 2025. 64

130 DIZDAREVIC, J.; MICHALKE, M.; JUKAN, A. Engineering and experimentally benchmarking open source mqtt broker implementations. *arXiv preprint arXiv:2305.13893*, 2023. Disponível em: <https://arxiv.org/abs/2305.13893>. 65

131 TRIPATHI, P.; MIRAZ, M. H.; JOSHI, S. Comparative analysis of mongodb and influxdb for time series data management in iot environments: A study on performance, scalability, and concurrency. In: *2023 International Conference on Computing, Networking, Telecommunications Engineering Sciences Applications (CoNTESA)*. [S.l.: s.n.], 2023. p. 39–42. 65

132 SAEED, M. S. et al. Detection of non-technical losses in power utilities—a comprehensive systematic review. *Energies*, v. 13, n. 18, 2020. ISSN 1996-1073. Disponível em: <https://www.mdpi.com/1996-1073/13/18/4727>. 69