UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE MAGNO OLIVEIRA SILVA

USO DE MACHINE LEARNING PARA ANÁLISE DE SUBPERFORMANCE DE USINAS SOLARES

FORTALEZA

2026

ALEXANDRE MAGNO OLIVEIRA SILVA

USO DE MACHINE LEARNING PARA ANÁLISE DE SUBPERFORMANCE DE USINAS SOLARES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Fernando Luiz Marcelo Antunes.
Coorientador: Prof. Dr. Menaouar Berrehil El Katel.

FORTALEZA

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S578u Silva, Alexandre Magno Oliveira.
Uso de machine learning para análise de subperformance de usinas solares / Alexandre Magno Oliveira Silva. – 2026.
83 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2026.
Orientação: Prof. Dr. Fernando Luiz Marcelo Antunes.
Coorientação: Prof. Dr. Menaouar Berrehil El Katel.

1. Usinas fotovoltaicas. 2. Machine learning. I. Título.

CDD 621.3

ALEXANDRE MAGNO OLIVEIRA SILVA

USO DE MACHINE LEARNING PARA ANÁLISE DE SUBPERFORMANCE DE USINAS SOLARES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Aprovada em: 12/01/2026

BANCA EXAMINADORA

Prof. Dr. Fernando Luiz Marcelo Antunes (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Luiz Henrique Silva Colado Barreto
Universidade Federal do Ceará (UFC)

M.Engª. Juliana Carvalho de Alencar
Delfos Energy

Eng. Lucca Lemos Costa Guerra
Delfos Energy

AGRADECIMENTOS

A Deus, por ter me guiado ao longo desta jornada de aprendizado e pesquisa, conduzindo-me por este caminho e fortalecendo-me nos momentos desafiadores.

Aos meus pais, Juarez Corrêa e Ana Angelica, por todo o apoio ao longo dessa caminhada, por acreditarem em mim e pelo incentivo constante desde o início da minha trajetória. Pessoas de extrema garra e resiliência, que me educaram, me ensinaram valores fundamentais e me mostraram como viver a vida com responsabilidade e perseverança.

Aos meus avós, João Alexandre e Nilza, por serem grandes exemplos de vida para mim e por cuidarem de mim, mesmo lá de cima. Gostaria que vocês pudessem ver esta conquista de perto, mas sei que estão orgulhosos ao ver tudo o que consegui.

À minha irmã Juliana, por sempre ser uma mulher extremamente guerreira, que me mostrou que devemos acreditar para alcançar nossos objetivos; e ao meu irmão Vinicius, por ter cuidado de mim quando pequeno, por estar ao meu lado em todas as brincadeiras e por sempre tornar o ambiente mais alegre, mesmo diante das dificuldades que a vida nos impôs. Ao agradecê-los, não posso deixar de mencionar os presentes que vieram através deles: Jéssica, esposa do meu irmão, e meus dois sobrinhos, Luiz e Miguel. Luiz, que está sempre comigo no dia a dia, é um dos meus principais apoios e, mesmo sendo tão novo, já demonstra ser um garoto forte, do qual tenho absoluta certeza de que se tornará um grande homem. Miguel, de quem tenho absoluta certeza de que será um grande artista, é um menino incrível, com um futuro promissor pela frente. À minha tia Joilza, uma pessoa incrível, que sempre me ensinou a enxergar o lado bom da vida e a ver tudo de maneira mais tranquila, sendo alguém simplesmente maravilhosa.

À minha namorada, Lindaiane, que vem me apoiando ao longo desses quase dois anos, sempre me ajudando a superar minhas ansiedades e desafios, estando ao meu lado em todos os momentos. Por tornar tudo mais bonito e agradável, por pensar sempre nos melhores planos e por permanecer comigo tanto nos momentos bons quanto nos momentos difíceis que enfrentei ao longo dessa jornada. Estendo também meus agradecimentos à sua família, que me acolheu com carinho e respeito. À Lindalva, Júlio, Emerson, Carol, Elderson, Thalia e Bento, pelas conversas, pelo incentivo e pela alegria que tornam os momentos em família ainda mais especiais.

Ao meu grande amigo, Victor Bruno, que sempre foi um irmão mais velho para mim, inspirando-me em diversas áreas e oferecendo conselhos valiosos que me ajudaram a trilhar um ótimo caminho e a tomar várias decisões importantes, inclusive na escolha das áreas de estudo e de atuação profissional. Além disso, agradeço pelas pessoas que ele trouxe à minha

vida: sua esposa, Micayle, sempre alegre e uma luz em sua trajetória, e ao grande Ethan Elliot, extremamente inteligente e sempre sorridente.

Ao meu padrinho e à minha madrinha, Ricardo Medina e Ana Medina, que me ajudaram desde o meu nascimento, estando presentes em momentos marcantes da minha vida, sempre me apoiando quando precisei e contribuindo de forma significativa para a minha jornada.

A dois grandes amigos que fiz nessa trajetória universitária, Ian Wanderley e Rogério Neto, ao Ian por me fazer rir diante de tudo, pelas caronas voltando da faculdade e pela grande viagem que fizemos para Alemanha, e ao Rogério por sempre estar comigo desde o início da faculdade, fazendo todas as disciplinas comigo, e correndo atrás dos ônibus ao meu lado ao sair da faculdade.

Não poderiam faltar os grandes amigos do trabalho. Primeiramente, à minha líder, Juliana Alencar, que sempre apoiou minhas ideias e me proporcionou muitas oportunidades, às quais sou profundamente grato. Ao Alberto Albuquerque, diretor que sempre me inspira a seguir em frente, mesmo com as ideias mais ousadas, e que, acima de tudo, sempre me ajudou imensamente. Ao meu time, Rhuan, Larah, Lucca e Michele, que também fizeram parte dessa jornada, tornando meu dia a dia mais agradável. Não posso esquecer da Nathianne, que, logo no surgimento da ideia deste TCC, sentou-se ao meu lado e me deu diversas orientações sobre como prosseguir, estruturar o trabalho e definir os melhores passos a seguir. A todos vocês, meu sincero agradecimento, pois uma parte significativa do conhecimento aplicado neste trabalho foi adquirida graças aos ensinamentos e ao treinamento que recebi de cada um.

E também aos vários amigos que fiz durante a graduação, em especial ao meu grande grupo de amigos Victor Silvestre, André Pina, Douglas, Eduardo Vilas Boas, Enzo, Igor Hammom, Luan, Renan Sá e Vitor Almeida, que sempre me ajudaram ao longo da graduação, seja nos trabalhos em equipe, nas revisões pré-provas ou em tantos outros momentos. Incluo aqui também a Mariana e meus amigos da Psicologia, que ao longo desse período muito me ensinaram e contribuíram para essa caminhada.

Ao meu orientador, Fernando Antunes, pela orientação ao longo da graduação, pelas conversas e direcionamentos que contribuíram para minha formação acadêmica e pelas experiências compartilhadas, especialmente durante a viagem para a Alemanha, que foram importantes para meu crescimento pessoal e profissional.

"Se eu fui capaz de ver mais longe, foi por estar sobre o ombro de gigantes." (Isaac Newton)

RESUMO

A expansão da energia solar em larga escala impõe desafios crescentes para a operação e manutenção (O&M), uma vez que métodos tradicionais de monitoramento baseados em limites fixos *Supervisory Control and Data Acquisition* (SCADA) muitas vezes falham na identificação de anomalias sutis. Este trabalho apresenta um sistema automatizado para detecção de falhas em *stringboxes* de usinas solares *utility-scale* utilizando técnicas de aprendizado de máquina. A abordagem propõe uma arquitetura *One-vs-All* baseada no algoritmo *eXtreme Gradient Boosting* (XGBoost) para identificar três classes distintas de anomalias: *stringboxes* zeradas (perda total de comunicação ou potência), subperformance (baixa eficiência contínua) e sombreamento (distorções temporais na curva de geração). A metodologia utilizou dados reais de uma usina no Nordeste do Brasil, aplicando uma engenharia de atributos que extraiu 28 características estatísticas e morfológicas das séries temporais diárias. Os resultados validaram a eficácia do sistema: o detector de *stringboxes* zeradas alcançou desempenho ideal (100% de F1-Score); o detector de subperformance obteve 95,22% de acurácia e F1-Score de 0,7059, demonstrando robustez contra falsos positivos; e o detector de sombreamento, validado via *k-fold* estratificado, atingiu F1-Score de 0,6368, comprovando a capacidade de capturar padrões temporais mesmo em cenários de dados desbalanceados. O sistema demonstra ser uma ferramenta viável para auxiliar a tomada de decisão em O&M, superando limitações de abordagens puramente determinísticas.

Palavras-chave: energia solar; machine learning; detecção de falhas; XGBoost; usinas fotovoltaicas; manutenção preditiva.

ABSTRACT

The expansion of large-scale solar power generation imposes increasing challenges for operation and maintenance (O&M), as traditional monitoring methods based on fixed thresholds (SCADA) often fail to identify subtle anomalies. This work presents an automated fault detection system for stringboxes in utility-scale solar plants using machine learning techniques. The approach proposes a One-vs-All architecture based on the XGBoost algorithm to identify three distinct anomaly classes: zero-output stringboxes (total loss of communication or power), underperformance (continuous low efficiency), and shading (temporal distortions in the generation curve). The methodology used real data from a plant in Northeast Brazil, applying feature engineering that extracted 28 statistical and morphological features from daily time series. The results validated the system's effectiveness: the zero-output detector achieved ideal performance (100% F1-Score); the underperformance detector obtained 95.22% accuracy and an F1-Score of 0.7059, demonstrating robustness against false positives; and the shading detector, validated via stratified k-fold, reached an F1-Score of 0.6368, proving the ability to capture temporal patterns even in imbalanced data scenarios. The system proves to be a viable tool to support O&M decision-making, overcoming the limitations of purely deterministic approaches.

Keywords: solar energy; machine learning; fault detection; XGBoost; photovoltaic plants; predictive maintenance.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
FN	*Falso Negativo*
FP	*Falso Positivo*
GHI	*Global Horizontal Irradiance*
IRENA	*International Renewable Energy Agency*
ML	*Machine Learning*
MPPT	*Maximum Power Point Tracking*
POA	*Plane of Array*
PR	*Performance Ratio*
SCADA	*Supervisory Control and Data Acquisition*
TN	*Verdadeiro Negativo*
TP	*Verdadeiro Positivo*
XGBoost	*eXtreme Gradient Boosting*

LISTA DE SÍMBOLOS

E_f	Energia produzida real (kWh)
E_r	Energia de referência teórica (kWh)
f_k	Função da k-ésima árvore de decisão (XGBoost)
G_{POA}	Irradiância medida no plano inclinado (W/m^2)
G_{ref}	Irradiância de referência (1000 W/m^2)
I_{sc}	Corrente de curto-circuito (A)
I_{string}	Corrente da string fotovoltaica (A)
K	Número total de árvores no modelo XGBoost
N	Número de módulos em série ou entradas
P_{AC}	Potência ativa de saída AC (kW)
P_{mp}	Potência no ponto de máxima potência (W)
P_{nom}	Potência nominal instalada DC ($P_{DC,rated}$)
P_{norm}	Potência ativa normalizada (p.u.)
P_{SB}	Potência total da stringbox (W)
T_{amb}	Temperatura ambiente ($^\circ C$)
T_{mod}	Temperatura de operação do módulo ($^\circ C$)
V_{DC}	Tensão do barramento de corrente contínua (V)
V_{oc}	Tensão de circuito aberto (V)
V_{string}	Tensão total da string fotovoltaica (V)
x_i	Vetor de dados de entrada (features)
y_i	Valor real (alvo) da variável
$\hat{y}_i$	Valor predito pelo modelo
β	Coeficiente de regressão linear
ε	Termo de erro ou resíduo estatístico
γ	Coeficiente de temperatura de potência ($\%/^\circ C$)
ρ	Albedo (coeficiente de reflexão do solo)
Ω	Termo de regularização (penalidade de complexidade)

SUMÁRIO

1 INTRODUÇÃO

O presente trabalho está estruturado em cinco capítulos, organizados de forma a conduzir o leitor desde a contextualização do problema até a validação da solução proposta. O Capítulo 1 apresenta a introdução, detalhando a justificativa, a motivação e os objetivos (geral e específicos) que nortearam o desenvolvimento da pesquisa.

O Capítulo 2 expõe a revisão bibliográfica e a fundamentação teórica, abordando os conceitos essenciais de sistemas fotovoltaicos e as limitações dos métodos tradicionais de monitoramento (SCADA). Além disso, discute os fundamentos de Aprendizado de Máquina e revisa os principais algoritmos utilizados para detecção de anomalias.

No Capítulo 3, é apresentada a metodologia, descrevendo o objeto de estudo, a base de dados utilizada e o processo de engenharia de atributos (*feature engineering*). Este capítulo detalha também a arquitetura do sistema proposto, baseada na estratégia *One-vs-All* com o algoritmo XGBoost, e define as métricas de avaliação.

O Capítulo 4 dedica-se à apresentação e discussão dos resultados. Nele, é analisado o desempenho global do sistema e a eficácia específica dos detectores de "Stringbox Zerada", "Subperformance" e "Sombreamento", complementado por uma análise qualitativa de casos reais.

Por fim, o Capítulo 5 apresenta as conclusões finais do estudo, sintetizando os principais achados e oferecendo sugestões de trabalhos futuros para a continuidade da pesquisa.

1.1 Justificativa e motivação

No século XIX, foi descoberto o efeito fotovoltaico (Becquerel, 1839), processo de conversão direta de energia luminosa em energia elétrica, cuja aplicação ainda não era viável economicamente à época. Por essa razão, as primeiras células solares comerciais surgiram apenas em 1954 (Chapin *et al.*, 1954), desenvolvidas pelo Bell Labs, sendo utilizadas em maior escala somente em satélites na década de 1960. Apenas décadas depois, tornaram-se competitivas para aplicação em larga escala.

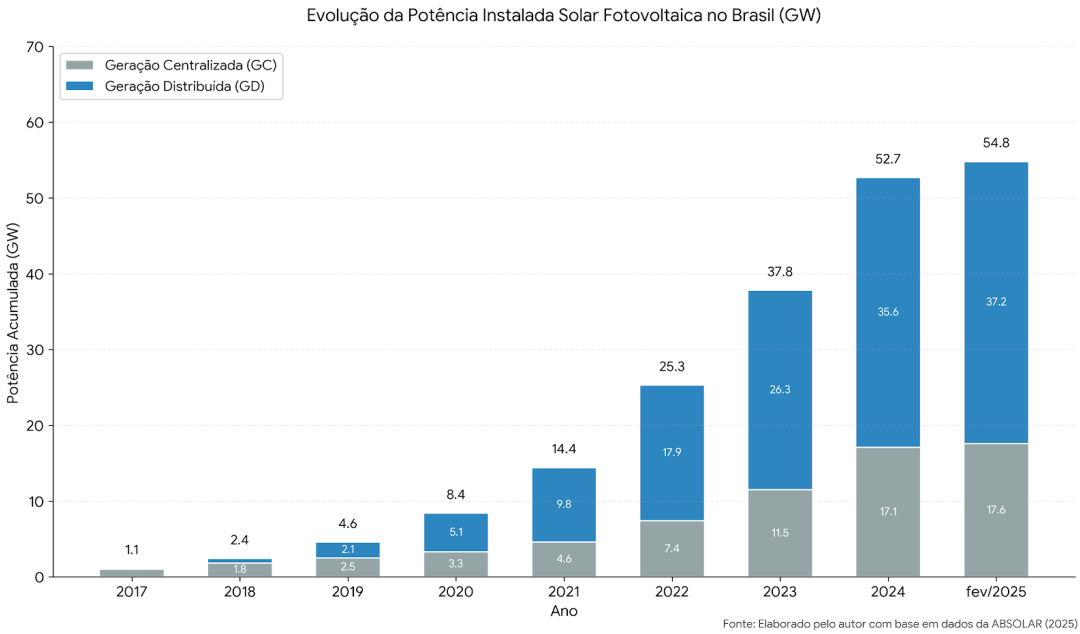
A energia solar fotovoltaica começou a se expandir rapidamente no início do século XXI, impulsionada pelo aumento global da demanda energética (Agency, 2020) e pela necessidade de fontes renováveis. Além disso, a queda drástica no custo dos módulos entre 2010 e 2020 (NREL, 2021) contribuiu significativamente para essa expansão. Como resultado, a energia

fotovoltaica se consolidou como uma das fontes que mais crescem no mundo, apresentando baixos custos de instalação e manutenção, além de alta relevância ambiental.

Nos últimos anos, o Brasil passou a ocupar uma posição de destaque no cenário global da energia solar fotovoltaica, conforme dados da *International Renewable Energy Agency* (IRENA) (IRENA, 2024), que posicionam o país entre os maiores mercados solares do mundo. O setor apresenta taxas de crescimento superiores à média mundial, impulsionado por fatores regulatórios, econômicos e pela expansão da infraestrutura elétrica nacional.

A criação da Resolução Normativa Agência Nacional de Energia Elétrica (ANEEL) nº 482/2012 (ANEEL, 2012) foi um marco para o setor solar brasileiro, ao estabelecer o Sistema de Compensação de Energia Elétrica (*net metering*) e permitir a venda de energia excedente para a rede. A partir dessa regulamentação, observou-se um crescimento exponencial da geração distribuída no Brasil, com milhões de unidades consumidoras aderindo à tecnologia fotovoltaica (ABSOLAR, 2025). Paralelamente, o segmento de geração centralizada também apresentou forte expansão, impulsionado pelos leilões de energia realizados ao longo da última década. A Figura 1 ilustra a evolução de potência instalada no Brasil, detalhando a participação por tipo de empreendimento solar - Geração Distribuída (GD) e Geração Centralizada (GC).

Figura 1 – Evolução da potência instalada solar no Brasil.



Fonte: Elaborado pelo autor com base em dados da ABSOLAR (ABSOLAR, 2025).

Assim, com o aumento da escala da geração fotovoltaica, tanto em quantidade de usinas centralizadas quanto em usinas distribuídas, a operação e manutenção desses sistemas

tornou-se cada vez mais complexa. Usinas atuais chegam a ter dezenas de milhares de módulos e mais de mil inversores, gerando grandes volumes de dados operacionais. Nesse contexto, pequenas falhas como sombreamentos parciais, degradações anômalas ou paradas de *string*, podem resultar em perdas significativas de energia ao longo do tempo. E com isso, conforme a quantidade de dados gerados aumenta, métodos tradicionais de análises de dados acabam por serem insuficientes para acompanhar a velocidade e o volume de geração de dados, dificultando a detecção precoce de falhas e de tomadas de decisão eficientes.

Além do grande volume de informações gerado continuamente pelos equipamentos fotovoltaicos, a complexidade da interpretação desses dados representa um desafio adicional. As variáveis elétricas e ambientais são registradas em alta frequência por inversores, *stringboxes* e estações meteorológicas, resultando em milhões de pontos de medição ao longo de poucos dias de operação. Além disso, segundo (Villalva; Gazoli, 2015), fatores externos como variabilidade da irradiância, temperatura ambiente, sombreamentos temporários e degradação natural dos módulos alteram o comportamento não-linear da potência gerada, tornando difícil distinguir entre anomalias reais e variações normais do sistema. Esse conjunto de características torna a etapa de diagnóstico especialmente desafiadora, sobretudo em usinas de grande porte com centenas de inversores e milhares de *strings*.

Tradicionalmente, a análise de performance em sistemas fotovoltaicos é conduzida por meio de softwares SCADA (*Supervisory Control and Data Acquisition*) e *dashboards* operacionais. Embora fundamentais para a operação em tempo real, essas ferramentas dependem fortemente de interpretação humana e de alarmes configurados manualmente baseados em limiares fixos (Pinho; Galdino, 2014). Na prática, muitas falhas relevantes não geram alarmes diretos — como sombreamentos intermitentes, degradação acelerada de módulos ou *strings* operando abaixo da referência — fazendo com que parte significativa das perdas permaneça oculta ou seja identificada tardiamente. Em resposta a essas limitações, plataformas de monitoramento avançado passaram a incorporar técnicas de análise de dados massivos para permitir uma avaliação operacional mais abrangente.

Com o avanço das tecnologias de processamento, métodos de inteligência artificial têm se mostrado particularmente adequados para lidar com a estocacidade e o volume das informações geradas em sistemas de energias renováveis. Conforme destacam (Mellit; Kalogirou, 2008), diferentemente das técnicas tradicionais, algoritmos de *machine learning* são capazes de aprender a relação complexa entre as variáveis de entrada e identificar automaticamente padrões

anômalos, mesmo quando sutis. Modelos supervisionados podem ser treinados com histórico de falhas conhecidas, enquanto métodos não supervisionados permitem detectar desvios mesmo na ausência de rótulos prévios.

Diante desse cenário, torna-se evidente a necessidade de métodos que combinem escalabilidade, capacidade de generalização e autonomia na identificação de anomalias operacionais. A aplicação de técnicas de aprendizado de máquina em dados fotovoltaicos representa, portanto, uma oportunidade promissora para aprimorar o desempenho, reduzir perdas energéticas e apoiar a tomada de decisão em ambientes complexos. É nesse contexto que se insere o presente trabalho.

1.2 Objetivos

Diante dos desafios relacionados ao grande volume de dados, à variabilidade operacional e às limitações dos métodos tradicionais de monitoramento, os objetivos deste trabalho concentram-se no desenvolvimento e avaliação de uma abordagem baseada em aprendizado de máquina para detecção automática de anomalias em *stringboxes* fotovoltaicas.

1.2.1 Objetivo geral

Desenvolver, implementar e avaliar um método baseado em aprendizado de máquina capaz de identificar automaticamente anomalias operacionais em *stringboxes* fotovoltaicas, utilizando dados históricos de potência, irradiância e referência.

1.2.2 Objetivos específicos

- Realizar o tratamento, seleção e preparação dos dados de operação das *stringboxes*;
- Investigar diferentes técnicas de aprendizado de máquina adequadas ao problema de detecção de anomalias;
- Treinar e validar modelos supervisionados e/ou não supervisionados para identificação de desvios operacionais;
- Comparar o desempenho entre diferentes modelos e estratégias de detecção;
- Avaliar a capacidade do método proposto de identificar falhas sutis ou não triviais, como sombreamento parcial, degradação e perdas em *strings*;
- Discutir as limitações, potenciais aplicações e perspectivas de uso da abordagem desenvol-

vida em ambientes reais de operação.

2 REVISÃO BIBLIOGRÁFICA E FUNDAMENTAÇÃO TEÓRICA

A fim de contextualizar o desenvolvimento da abordagem proposta neste trabalho, esta seção apresenta uma revisão dos principais conceitos relacionados ao monitoramento de sistemas fotovoltaicos, técnicas de análise de dados e métodos de aprendizado de máquina aplicados à detecção de anomalias. A revisão é estruturada de forma hierárquica, iniciando pelos fundamentos gerais de operação e desempenho de usinas solares, avançando para métodos tradicionais de diagnóstico e, por fim, discutindo modelos de machine learning relevantes para o problema tratado.

2.1 Fundamentos de desempenho em sistemas fotovoltaicos

O primeiro passo para compreender as anomalias operacionais em sistemas fotovoltaicos é a caracterização dos fatores que determinam o desempenho de um gerador solar. Variáveis como irradiância, temperatura dos módulos e condições de operação elétrica influenciam diretamente a potência entregue ao inversor, sendo fundamentais para qualquer abordagem de diagnóstico.

2.1.1 Recursos Solarimétricos

2.1.1.1 Irradiância Solar

A irradiância é definida como a taxa de energia radiante incidente por unidade de área, expressa em Watts por metro quadrado (W/m^2). Segundo (Villalva; Gazoli, 2015), esta é a variável instantânea que determina a potência elétrica gerada pelos módulos fotovoltaicos em um dado momento. Variações bruscas na irradiância, causadas pela passagem de nuvens, impactam diretamente a corrente de saída do gerador, sendo o principal fator de variabilidade na geração.

2.1.1.2 Irradiância Global Horizontal (GHI)

A Irradiância Global Horizontal (*Global Horizontal Irradiance* (GHI)) representa a energia total incidente em uma superfície horizontal terrestre. Conforme definido pelo manual do CEPEL (Pinho; Galdino, 2014), ela é composta pela radiação direta (vinda do disco solar) e pela radiação difusa (espalhada pela atmosfera). Esta é a variável padrão medida por piranômetros instalados horizontalmente em estações solarimétricas para fins de monitoramento meteorológico.

2.1.1.3 Albedo e Componente Refletida

O Albedo (ρ) é um coeficiente adimensional que representa a refletividade da superfície do solo, variando de 0 (absorção total) a 1 (reflexão total). Superfícies claras e polidas apresentam alto albedo, enquanto superfícies escuras e rugosas apresentam baixo albedo.

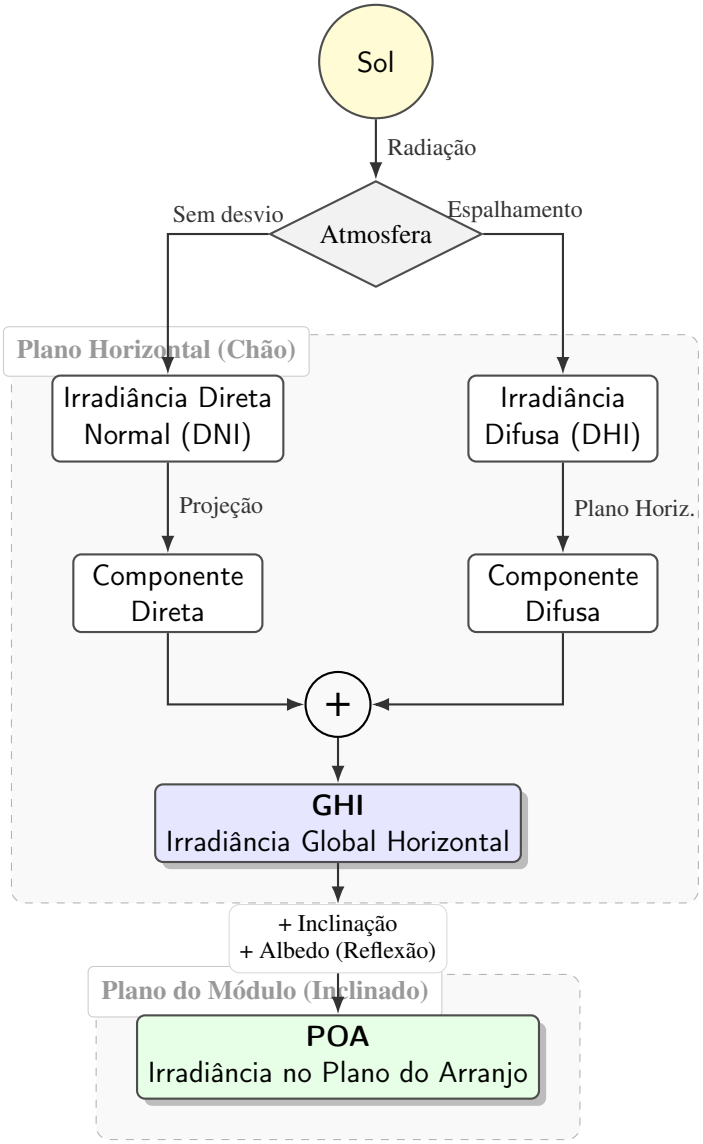
Segundo (Villalva; Gazoli, 2015), a radiação refletida pelo solo (*ground reflected radiation*) torna-se uma componente significativa para superfícies inclinadas. Em usinas solares típicas, onde o solo é coberto por grama ou brita, o albedo usualmente varia entre 0,15 e 0,25. Esta componente contribui para a irradiância total que atinge o plano do arranjo, sendo proporcional à GHI e ao fator de visão do módulo em relação ao solo.

2.1.1.4 Irradiância no Plano do Arranjo (POA)

A Irradiância no Plano do Arranjo (*Plane of Array* (POA)) refere-se à radiação incidente na superfície inclinada dos módulos fotovoltaicos. Diferentemente da GHI, a POA considera a inclinação e a orientação azimutal dos painéis, sendo a soma vetorial de três componentes: a direta, a difusa e a refletida pelo solo (calculada através do albedo) (Villalva; Gazoli, 2015).

Para fins de modelagem de desempenho e cálculo da potência de referência, a POA é a métrica mais precisa, pois apresenta a maior correlação linear com a potência de saída do inversor, representando a energia efetivamente disponível para conversão fotovoltaica. A Figura 2 ilustra de forma esquemática a decomposição das componentes da irradiância que resultam na POA.

Figura 2 – Fluxograma das componentes da irradiância solar: da emissão até a incidência no plano inclinado.



Fonte: Elaboração do autor.

2.1.2 Temperatura do Módulo

A temperatura das células fotovoltaicas exerce influência direta sobre o desempenho elétrico dos módulos. Segundo (Villalva; Gazoli, 2015), o aumento da temperatura reduz a tensão de operação e, consequentemente, a ponto de máxima potência (P_{mp}). Esse efeito térmico ocorre principalmente devido à diminuição da *bandgap* do material semicondutor com o aquecimento. Assim, para condições de irradiância semelhantes, módulos mais aquecidos tendem a apresentar menor produção de energia. Apesar de sua relevância para modelagem detalhada e estimativas precisas de potência, muitos sistemas operacionais não registram a temperatura do módulo diretamente, utilizando apenas a temperatura ambiente como aproximação. No contexto deste

trabalho, a temperatura não é utilizada como variável de entrada nos modelos, mas permanece como um fator importante na compreensão do comportamento físico da geração fotovoltaica.

2.1.3 Performance Ratio (PR)

O *Performance Ratio* (PR) é um dos principais indicadores utilizados para quantificar o impacto agregado das perdas na operação de sistemas fotovoltaicos (Pinho; Galdino, 2014). Ele expressa a razão entre a energia efetivamente produzida pelo sistema (E_f) e a energia teórica de referência (E_r), definida com base na irradiância incidente no plano do arranjo.

Para fins de diagnóstico operacional e análise de desempenho instantâneo, o PR pode ser calculado a partir de variáveis de potência e irradiância medidas em tempo real. A formulação utilizada neste trabalho é apresentada na Equação 2.1:

$$PR = \frac{E_f}{E_r} = \frac{\left(\frac{P_{AC}}{P_{nom}}\right)}{\left(\frac{G_{POA}}{G_{ref}}\right)} \tag{2.1}$$

Em que:

- P_{AC}: Potência ativa de saída dos inversores (kW);
- P_{nom}: Potência nominal instalada em corrente contínua nas condições padrão (kWp);
- G_{POA}: Irradiância medida no plano do arranjo (W/m²);
- G_{ref}: Irradiância de referência em condições padrão, fixada em 1000 W/m².

Nesta formulação, o termo $\left(G_{POA}/G_{ref}\right)$ representa o recurso solar disponível no instante analisado, enquanto (P_{AC}/P_{nom}) expressa o fator de carga instantâneo da usina. Dessa forma, valores reduzidos de PR indicam que a potência entregue está inferior ao esperado para o nível de irradiância incidente, revelando possíveis perdas anômalas ou falhas operacionais.

2.1.4 Séries Fotovoltaicas (Strings)

Uma série fotovoltaica, comumente denominada pelo termo técnico em inglês *string*, consiste na associação elétrica de múltiplos módulos fotovoltaicos conectados em série. O objetivo principal desta configuração é elevar a tensão de saída em corrente contínua (CC) para atingir os níveis operacionais exigidos pela janela de rastreamento de máxima potência (*MPPT*) do inversor (Villalva; Gazoli, 2015).

A Figura 3 ilustra o esquema elétrico desta conexão. Do ponto de vista de circuitos, a tensão total do arranjo é a soma das tensões individuais dos módulos, enquanto a corrente

elétrica que percorre a string é idêntica para todos os componentes associados. As relações fundamentais são dadas pelas Equações 2.2 e 2.3:

$$V_{string} = \sum_{i=1}^{N} V_{mod,i} \tag{2.2}$$

$$I_{string} = I_{mod} \tag{2.3}$$

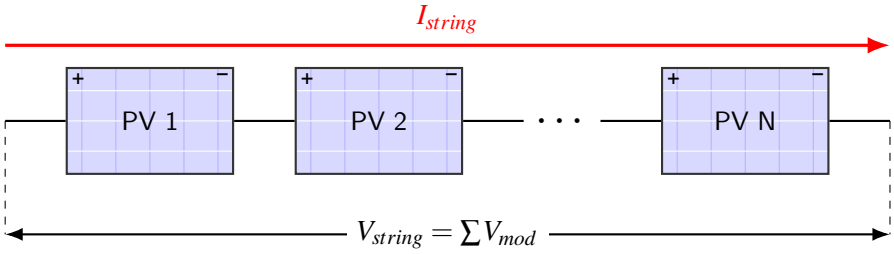
Em que:

- V_{string}: Tensão total da *string* fotovoltaica (V);
- N: Número total de módulos conectados em série;
- $V_{mod,i}$: Tensão individual do i-ésimo módulo da série (V);
- I_{string}: Corrente elétrica total que percorre a *string* (A);
- I_{mod}: Corrente elétrica que percorre os módulos individuais (A).

Esta característica elétrica implica que o desempenho de uma *string* é limitado pelo seu "elo mais fraco". Caso um único módulo sofra sombreamento, sujeira ou defeito interno, a corrente de toda a série será restringida pela capacidade desse módulo degradado, resultando em perdas desproporcionais de potência conhecidas como perdas por *mismatch* (descasamento) (Pinho; Galdino, 2014).

No contexto de usinas de grande porte, as strings constituem a menor unidade de geração, embora o monitoramento frequentemente ocorra de forma agregada nas caixas de junção.

Figura 3 – Representação esquemática de uma string fotovoltaica: a corrente é comum a todos os módulos, enquanto as tensões se somam.



Fonte: Elaboração do autor.

2.1.5 Topologias de Inversores e Arquitetura de Sistema

O inversor fotovoltaico é o dispositivo de eletrônica de potência responsável pela conversão da corrente contínua (CC) gerada pelos módulos em corrente alternada (CA) compatí-

vel com a rede elétrica, além de realizar o rastreamento do ponto de máxima potência (*Maximum Power Point Tracking* (MPPT)).

A escolha da topologia do inversor define a arquitetura de monitoramento da usina e a necessidade de equipamentos auxiliares. Conforme ilustrado na **Figura 4**, destacam-se duas configurações principais em usinas de grande porte:

2.1.5.1 Inversores Centrais

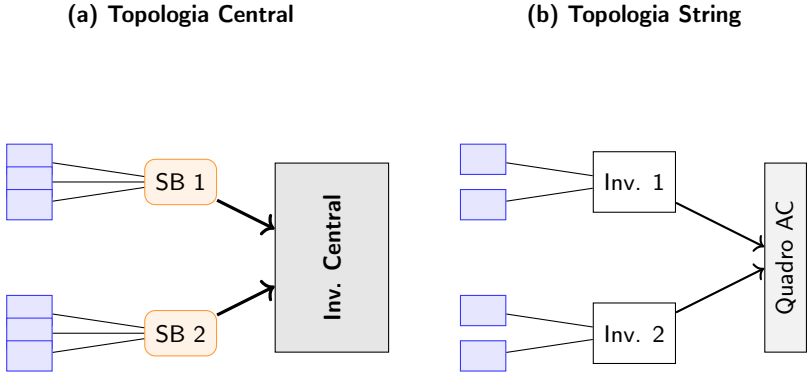
Nesta configuração, apresentada na **Figura 4(a)**, um único inversor de alta potência (tipicamente > 1 MW) processa a energia de milhares de módulos.

Devido à inviabilidade física de conectar milhares de cabos diretamente à entrada do inversor, torna-se mandatória a utilização de Caixas de Junção (*Stringboxes*) distribuídas pelo campo. Estes equipamentos realizam o paralelismo das *strings* e enviam a energia ao inversor através de um único par de cabos de alimentação (*DC Feeder*) (Villalva; Gazoli, 2015). Esta é a topologia foco deste trabalho, onde o monitoramento granular depende inteiramente da inteligência embarcada nas *stringboxes*.

2.1.5.2 Inversores de String (Descentralizados)

Nesta topologia, ilustrada na **Figura 4(b)**, múltiplos inversores de menor potência são instalados próximos aos arranjos. Cada inversor recebe diretamente um pequeno número de *strings*, eliminando frequentemente a necessidade de caixas de junção externas (Pinho; Galdino, 2014). Embora ofereçam maior granularidade nativa, a topologia central ainda é predominante em usinas *utility-scale* devido à robustez e facilidade de manutenção centralizada.

Figura 4 – Comparativo de topologias: (a) Inversor Central, que exige o uso de Stringboxes (SB) para concentrar as séries; (b) Inversor String, onde as séries são conectadas diretamente ao equipamento.



Fonte: Elaboração do autor.

2.1.6 Stringboxes: Caixas de Junção para Agrupamento de Strings

As *Stringboxes* desempenham um papel estrutural fundamental na topologia de usinas fotovoltaicas, sendo componentes característicos e indispensáveis em sistemas baseados em **Inversores Centrais**. Devido à inviabilidade física de conectar milhares de cabos diretamente à entrada de um único inversor de alta potência, as stringboxes realizam a função de agrupar os circuitos de campo em estágios intermediários (Villalva; Gazoli, 2015).

Conforme detalhado no diagrama funcional da Figura 5, a função elétrica deste equipamento é realizar o paralelismo das N strings conectadas em suas entradas, entregando ao inversor central uma corrente total (I_{total}) através de um único par de cabos de saída de maior bitola (*DC Feeder*).

Do ponto de vista de monitoramento, a stringbox representa a granularidade mínima de aquisição de dados deste estudo. O sinal de interesse não é a corrente individual de cada série, mas sim a Potência Total da Stringbox (P_{SB}), calculada pelo produto da tensão do barramento (V_{DC}) pela corrente total agregada:

$$P_{SB}(t) = V_{DC}(t) \cdot \sum_{i=1}^{N} I_{string,i}(t) \tag{2.4}$$

Nesta abordagem, falhas internas — como a desconexão de uma única string ou sombreamento parcial em um subconjunto de módulos — manifestam-se como perturbações na curva de potência total da stringbox. O desafio do sistema de detecção de anomalias reside em distinguir essas perdas parciais das variações naturais de irradiância, tarefa que se torna mais complexa devido ao mascaramento causado pela agregação de correntes no barramento comum.

2.1.6.1 Potência Normalizada

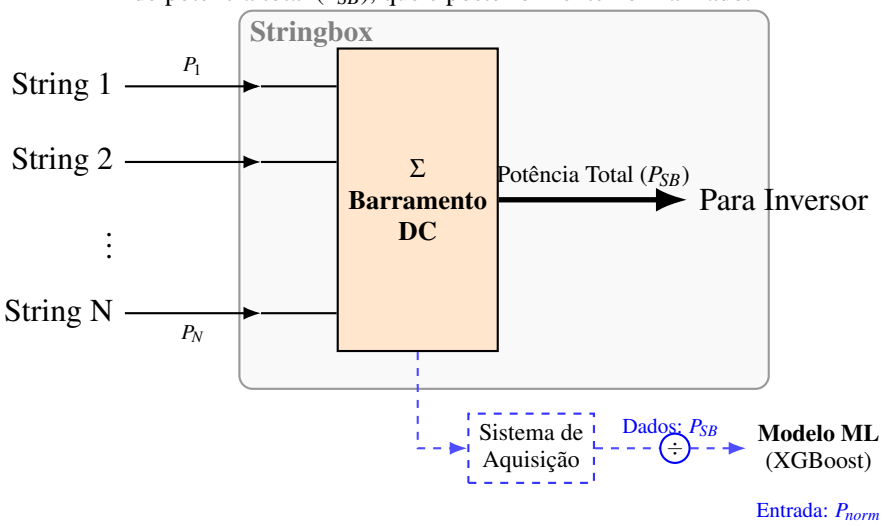
Para permitir a comparação direta entre stringboxes com diferentes capacidades instaladas e alimentar os modelos de aprendizado de máquina com dados padronizados, utiliza-se o conceito de Potência Normalizada (P_{norm}).

A normalização remove a dimensão de "tamanho"do equipamento, transformando a potência absoluta (Watts) em uma grandeza relativa. Conforme fluxo de dados apresentado na parte inferior da Figura 5, o sinal bruto é dividido pela capacidade nominal instalada da stringbox ($P_{nom,SB}$):

$$P_{norm} = \frac{P_{SB}}{P_{nom,SB}} \tag{2.5}$$

Desta forma, uma P_{norm} de 1,0 indica operação em potência máxima nominal, enquanto valores próximos a 0 indicam falha total (stringbox zerada).

Figura 5 – Fluxo de potência e dados em uma Stringbox: múltiplas strings são agregadas para gerar um único sinal de potência total (P_{SB}), que é posteriormente normalizado.



Fonte: Elaboração do autor.

2.1.7 Sistemas de Rastreamento Solar (Trackers)

Para maximizar a captação de energia, grandes usinas fotovoltaicas frequentemente utilizam estruturas móveis denominadas rastreadores solares ou *trackers*. A função primordial destes dispositivos é orientar a superfície dos módulos fotovoltaicos de forma a minimizar o ângulo de incidência dos raios solares, mantendo-os o mais perpendicular possível à radiação direta durante todo o período diurno (Villalva; Gazoli, 2015).

Enquanto estruturas fixas possuem um ângulo de inclinação (*tilt*) e orientação azimutal constantes, os rastreadores movem-se autonomamente seguindo a trajetória do sol. Segundo (Pinho; Galdino, 2014), o uso de rastreadores pode incrementar a produção de energia entre 15% e 25% em comparação a sistemas fixos, dependendo da latitude local e das condições de nebulosidade.

No contexto de usinas de grande porte no Brasil, a tecnologia predominante é o Rastreador de Eixo Único Horizontal (*Horizontal Single-Axis Tracker* - HSAT). Nesta configuração, o eixo de rotação é alinhado na direção Norte-Sul, permitindo que os módulos girem de Leste (ao amanhecer) para Oeste (ao entardecer).

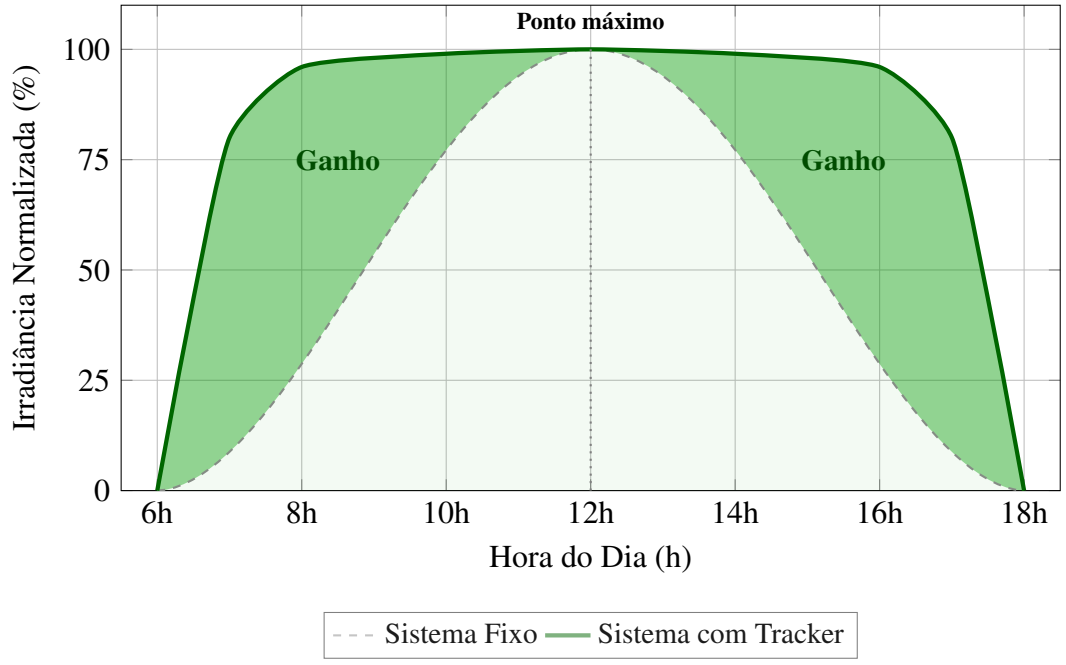
2.1.7.1 Impacto na Curva de Potência e Backtracking

A utilização de *trackers* altera substancialmente o perfil da curva de potência diária que serve de entrada para os modelos de aprendizado de máquina. Diferentemente da curva de "sino" típica de sistemas fixos, a curva de um sistema com rastreamento apresenta um perfil mais largo e achatado no topo, sustentando a potência máxima por um período prolongado.

Um aspecto crítico para a detecção de anomalias em sistemas com *trackers* é o algoritmo de *Backtracking* (retrocesso). No início da manhã e final da tarde, quando o sol está muito baixo no horizonte, o alinhamento perfeito com o sol faria com que uma fileira de módulos projetasse sombra sobre a fileira posterior (sombreamento mútuo). Para evitar esse efeito, o sistema de controle força os rastreadores a desviarem do ângulo ideal, "deitando"os módulos para evitar o bloqueio da luz na fileira vizinha (NREL, 2021).

A Figura 6 ilustra o ganho de irradiância proporcionado pelo rastreamento e o efeito do *backtracking* nas extremidades do dia. O modelo de detecção de falhas deve ser capaz de compreender que a redução de potência durante o *backtracking* é um comportamento intencional de proteção, e não uma anomalia de subperformance.

Figura 6 – Comparativo conceitual de irradiância (POA): Sistema Fixo vs. Rastreador (Tracker). Ambas atingem o mesmo pico ao meio-dia, mas o tracker apresenta uma curva mais larga ("ombros"), gerando ganho energético nas manhãs e tardes.



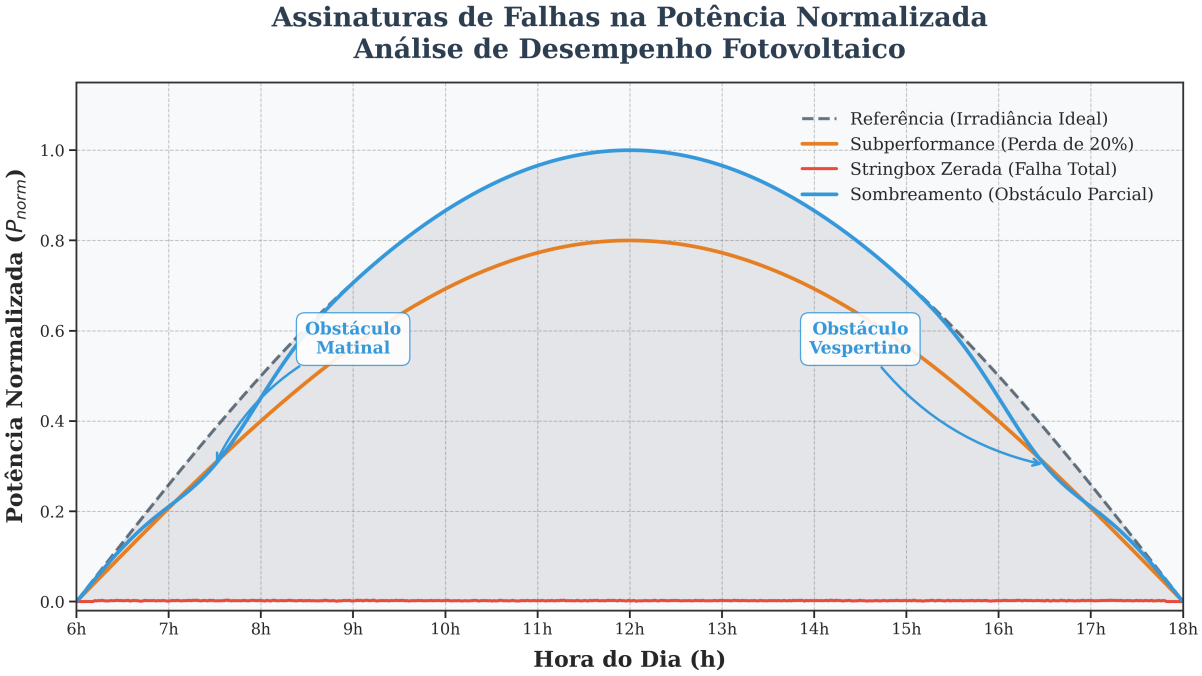
Fonte: Elaboração do autor.

2.1.8 Tipos de Falhas e Assinaturas de Dados

A operação de usinas fotovoltaicas está sujeita a anomalias que variam desde perdas catastróficas até degradações parciais. A visualização dessas falhas depende da tecnologia de montagem utilizada, uma vez que o perfil de referência muda drasticamente entre sistemas fixos e rastreadores.

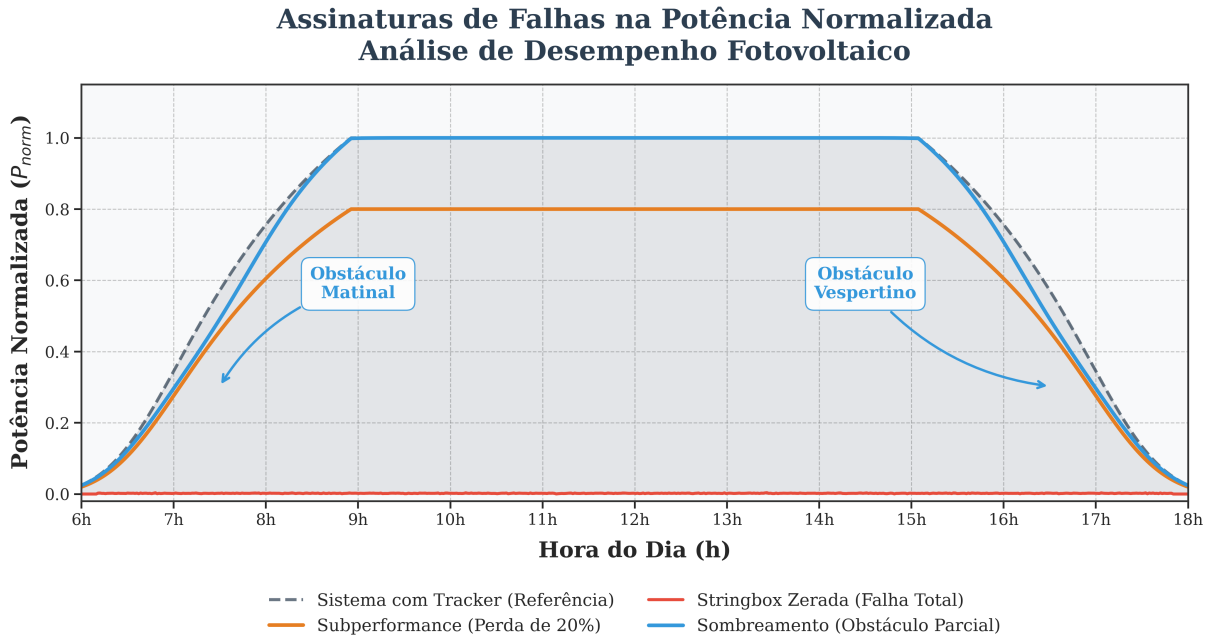
A Figura 7 apresenta as falhas sobre o perfil senoidal típico de estruturas fixas, enquanto a Figura 8 ilustra as mesmas condições sobre o perfil de "platô" característico de sistemas com *trackers*.

Figura 7 – Assinaturas características das falhas na curva de potência normalizada (Sistema Fixo). Note a deformação côncava causada pelo sombreamento (azul) em contraste com a perda constante da subperformance (laranja).



Fonte: Elaboração do autor.

Figura 8 – Assinaturas de falhas em sistema com Rastreador (Tracker). O perfil de "mesa" altera a morfologia do sombreamento, que tende a deformar os "ombros" da curva nas primeiras e últimas horas do dia.



Fonte: Elaboração do autor.

2.1.8.1 *Stringbox Zerada (Falha Total)*

Representada pela linha vermelha na base dos gráficos, esta condição caracteriza-se pela perda funcional da geração ($P_{norm} \approx 0$).

- **Causas Físicas:** Atuação de fusíveis gerais, abertura de chaves seccionadoras ou falhas no sistema de comunicação.
- **Assinatura:** A potência permanece nula ou apresenta apenas ruído residual de medição, independentemente da intensidade da irradiância incidente ou da tecnologia de rastreamento.

2.1.8.2 *Subperformance (String Fora)*

Ilustrada pela curva laranja, a subperformance denota uma operação contínua, porém com eficiência degradada.

- **Causas Físicas:** Desconexão ou queima de fusíveis de um subconjunto de *strings* (ex: perda de 2 strings em uma caixa de 10 entradas).
- **Assinatura:** A curva mantém a morfologia correta (seja senoidal ou quadrada) e alta correlação com a referência, porém com amplitude reduzida (*offset* negativo) ao longo de todo o período de geração.

2.1.8.3 Sombreamento Sistemático (Obstáculos Fixos)

Destacado em azul, este fenômeno difere da perda constante por apresentar dependência temporal e geométrica.

- **Causas Físicas:** Bloqueio da luz solar direta por obstáculos fixos (vegetação, edificações ou fileiras adjacentes) que ocorre apenas quando o sol se encontra em baixas elevações.
- **Assinatura:** Conforme os destaques "Obstáculo Matinal"e “Vespertino” nas Figuras 7 e 8, a anomalia manifesta-se como uma deformação côncava ("barriga") nas laterais da curva. Tanto no sistema fixo quanto no sistema com *tracker*, essa distorção ocorre predominantemente nas rampas de subida (manhã) e descida (tarde), momentos em que a elevação solar favorece a projeção de sombras. Nestes intervalos, a potência descola-se da referência ideal, retornando à normalidade (ou ao patamar de subperformance) quando o obstáculo deixa de bloquear a radiação direta. O desafio para o modelo de *machine learning* reside em diferenciar essa distorção não-linear da variação natural de irradiância.

2.1.9 Estação Solarimétrica e Instrumentação

Para garantir a confiabilidade do monitoramento de desempenho, usinas fotovoltaicas de grande porte são equipadas com estações solarimétricas locais. A especificação, instalação e manutenção destes instrumentos são regidas pela norma internacional IEC 61724-1 (IEC, 2017), que classifica os sistemas de monitoramento em classes de precisão (A, B ou C) conforme a incerteza admissível e a taxa de amostragem.

A disponibilidade de dados meteorológicos precisos e alinhados temporalmente com os dados elétricos é pré-requisito fundamental para a aplicação de algoritmos de aprendizado de máquina. Os principais instrumentos que compõem uma estação típica são:

2.1.9.1 Piranômetros e Células de Referência

O piranômetro é o instrumento padrão para medição da irradiância solar global. Segundo a IEC 61724-1, para sistemas de alta precisão, deve-se realizar a medição em dois planos distintos:

- **Irradiância Global Horizontal (GHI):** Medida por um piranômetro nivelado com o solo, utilizada para validação do recurso solar em relação a dados de satélite.
- **Irradiância no Plano do Arranjo (POA):** Medida por um piranômetro (ou célula de refe-

rência calibrada) instalado com a mesma inclinação e azimute dos módulos fotovoltaicos. Esta é a variável crítica (G_{POA}) que alimenta o cálculo da potência de referência e serve como *input* principal para o modelo XGBoost proposto neste trabalho, pois representa a energia efetivamente disponível para conversão (Villalva; Gazoli, 2015).

2.1.9.2 Sensores de Temperatura e Anemômetros

Além da irradiância, a estação monitora a temperatura ambiente (T_{amb}) e a temperatura da superfície do módulo (T_{mod}). A medição da temperatura do módulo é crucial, pois, conforme visto na Seção 2.1.2, a eficiência de conversão cai linearmente com o aquecimento. Em conformidade com a norma, sensores de temperatura (tipo RTD ou termopares) são fixados na parte posterior (*backsheet*) de módulos representativos para capturar o efeito térmico real sobre a geração.

2.2 Métodos tradicionais de monitoramento e diagnóstico

Tradicionalmente, o monitoramento de usinas fotovoltaicas baseia-se em sistemas SCADA e indicadores extraídos de dados elétricos e ambientais. Embora amplamente utilizados, esses métodos apresentam limitações importantes, sobretudo em cenários de grande escala, nos quais falhas sutis podem não gerar alarmes diretos ou podem permanecer encobertas por variabilidade natural da operação.

2.2.1 Sistemas Supervisórios (SCADA)

O SCADA constitui a espinha dorsal da operação de usinas fotovoltaicas de médio e grande porte. Trata-se de uma arquitetura de sistemas que integra *hardware* e *software* para permitir a aquisição de dados em tempo real, o controle remoto de dispositivos e o armazenamento histórico de variáveis operacionais (Pinho; Galdino, 2014).

Em uma planta solar típica, o SCADA atua interrogando ciclicamente os dispositivos de campo — inversores, *stringboxes*, rastreadores (*trackers*) e estações meteorológicas — por meio de protocolos industriais como Modbus TCP/RTU ou DNP3. Os dados coletados (tensão, corrente, potência, temperatura, códigos de erro) são centralizados em um servidor local e apresentados aos operadores através de uma Interface Homem-Máquina (IHM).

A Figura 9 apresenta uma arquitetura típica de um sistema SCADA aplicado a

usinas fotovoltaicas, destacando o fluxo de dados desde os sensores de campo até a camada de supervisão.

2.2.1.1 Lógica de Alarmes e Limitações

Apesar da eficiência na coleta e visualização, a capacidade de diagnóstico automatizado dos sistemas SCADA tradicionais é limitada. A detecção de falhas fundamenta-se, predominantemente, em regras determinísticas e limiares fixos (*thresholds*).

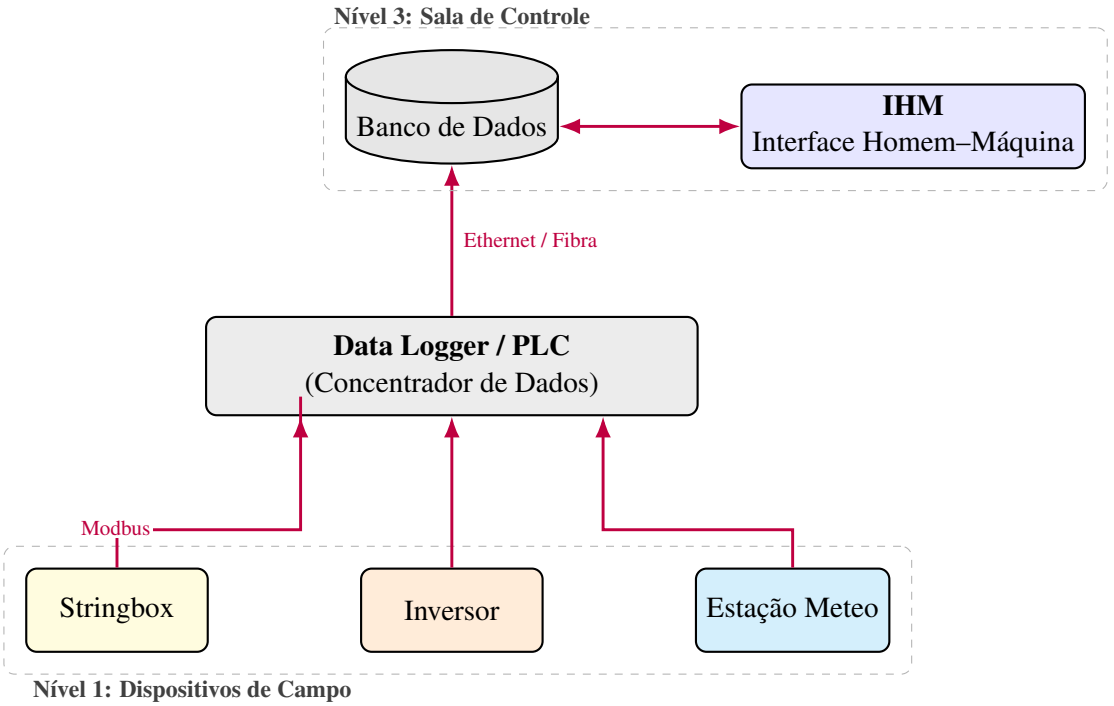
Um exemplo usual de regra SCADA pode ser representado como:

$$\text{SE } (P_{inversor} = 0) \text{ E } (GHI > 500\ W/m^2) \rightarrow \text{ALARME: Parada Inesperada} \tag{2.6}$$

Essa abordagem binária é eficaz para falhas severas, porém ineficiente para anomalias sutis. Conforme discutido por (Mellit; Kalogirou, 2008), a variabilidade natural da geração solar — como sombreamentos passageiros ou flutuações térmicas — dificulta o uso de limiares estáticos, resultando em:

1. **Falsos Positivos:** alarmes gerados por variações rápidas de irradiância;
2. **Falsos Negativos:** perdas reais (ex.: uma string desconectada) que não superam o limiar configurado.

Figura 9 – Arquitetura típica de um sistema SCADA em usina fotovoltaica: fluxo de dados desde os sensores de campo até a interface de operação.



Fonte: Elaboração do autor.

2.2.2 Dashboards Operacionais e o Mascaramento de Falhas

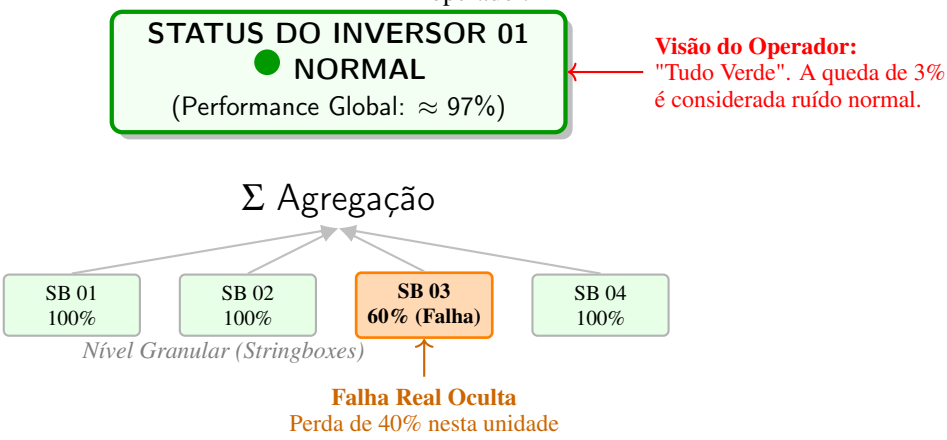
Os *dashboards* operacionais constituem a camada de visualização responsável por sintetizar o grande volume de dados provenientes do SCADA. Em sua forma mais simples — especialmente em sistemas cuja função é apenas apresentar indicadores agregados — esses painéis recorrem amplamente à consolidação de variáveis para facilitar a interpretação pelo operador.

Em plataformas dessa natureza, o status de centenas de *stringboxes* é frequentemente resumido em métricas globais por Inversor, por Setor ou para toda a usina (por exemplo: Potência Total, PR Médio, Energia Diária). Embora essa agregação seja útil para uma visão macro, ela pode introduzir um efeito de mascaramento de falhas, no qual anomalias localizadas são diluídas no indicador agregado.

Esse fenômeno não é uma limitação inerente a todos os sistemas de monitoramento modernos. Pelo contrário: plataformas avançadas, baseadas em *analytics* e modelos específicos como *digital twins*, implementam mecanismos dedicados justamente para evitar esse tipo de ocultamento. O mascaramento ocorre sobretudo em dashboards operacionais simplificados, ou em visualizações que dependem exclusivamente de média ou soma de grandezas elétricas.

A Figura 10 ilustra esse mecanismo. Se uma *stringbox* sofre subdesempenho severo (por exemplo, perda de 40%), mas ela compõe um conjunto maior de unidades agregadas, o impacto total pode ser reduzido a poucos pontos percentuais. O operador, ao visualizar apenas o indicador consolidado — tipicamente representado pela cor verde ou por um status "Normal" — pode não perceber a anomalia granular, a menos que realize uma investigação detalhada (*drill-down*).

Figura 10 – O fenômeno de mascaramento de falhas em *dashboards*: a subperformance severa de uma *stringbox* individual é diluída na agregação total do inversor, apresentando um status visual enganosamente positivo ao operador.



Fonte: Elaboração do autor.

2.2.3 Limitações da Abordagem Tradicional

Embora os sistemas SCADA e os *dashboards* operacionais sejam essenciais para a supervisão em tempo real, eles apresentam limitações importantes quando o objetivo é identificar falhas sutis ou perdas parciais de geração. De modo geral, esses sistemas dependem de regras fixas e da interpretação visual do operador, o que reduz sua sensibilidade a anomalias discretas.

2.2.3.1 Limitações de Limiar em um Ambiente Variável

A geração fotovoltaica varia constantemente por causa de nuvens, dispersão atmosférica e mudanças rápidas na irradiância. Quando o sistema utiliza limites fixos (*thresholds*) para disparar alarmes, surgem dois problemas típicos:

- **Limiar muito sensível:** gera muitos falsos alarmes em dias com nuvens rápidas, levando o operador a ignorá-los.
- **Limiar muito permissivo:** permite que perdas reais permaneçam escondidas, como uma stringbox produzindo 20% a menos, mas ainda dentro da faixa "aceitável" pelo SCADA

(Mellit; Kalogirou, 2008).

Ou seja, um valor fixo não consegue acompanhar um ambiente que muda minuto a minuto.

2.2.3.2 Sobrecarga Visual e Falta de Escalabilidade

Outra limitação relevante está relacionada ao volume de informações. Em uma usina fotovoltaica moderna, um operador pode precisar acompanhar simultaneamente centenas de inversores, milhares de stringboxes e dezenas de curvas, alarmes e indicadores operacionais. Segundo (Pinho; Galdino, 2014), esse cenário torna o processo de monitoramento predominantemente reativo, uma vez que a falha costuma ser percebida apenas após já ter causado impacto visível na operação. Além disso, o processo torna-se fortemente dependente do operador, pois profissionais diferentes podem interpretar os mesmos gráficos de maneiras distintas e, consequentemente, tomar decisões diferentes. Por fim, trata-se de um processo lento, já que a análise manual de curvas históricas pode demandar horas ou até dias de trabalho.

Esses fatores fazem com que falhas pequenas, mas frequentes, passem despercebidas — especialmente quando são diluídas na agregação por inversor ou por setor (ver Figura 10).

2.2.3.3 Comparativo entre as Abordagens

Para destacar as diferenças entre o método tradicional e a solução proposta, a Tabela1 resume as principais características de cada abordagem.

Tabela 1 – Comparativo entre métodos tradicionais e baseados em Machine Learning.

Característica	**Métodos Tradicionais (SCADA/ Regras)**	**Abordagem Proposta (Machine Learning/XGBoost)**
Lógica de Detecção	Regras fixas (Se... Então...).	Padrões aprendidos a partir dos dados.
Sensibilidade	Baixa para falhas parciais.	Alta — identifica desvios pequenos.
Adaptabilidade	Limiar estático.	Se ajusta às condições reais.
Interferência	Sofre com variações rápidas de irradiância.	Filtra ruídos e variações naturais.
Diagnóstico	Depende da inspeção manual.	Automático e escalável.

Fonte: Elaboração do autor.

2.3 Conceitos fundamentais de aprendizado de máquina

Técnicas de aprendizado de máquina têm se destacado como alternativas promissoras para o diagnóstico automático de sistemas fotovoltaicos, uma vez que permitem modelar relações complexas entre variáveis e identificar padrões anômalos de operação. Em termos gerais, algoritmos de *machine learning* podem ser classificados em métodos supervisionados e não supervisionados, cujas características diferem conforme a disponibilidade de rótulos e o tipo de tarefa a ser realizada.

2.3.1 Conceitos Fundamentais e Histórico

O Aprendizado de Máquina (*Machine Learning*) é um subcampo da inteligência artificial que se dedica ao estudo de algoritmos capazes de melhorar seu desempenho em tarefas específicas através da experiência. Uma das definições mais citadas é a de Arthur Samuel, que em 1959 descreveu o campo como "a área de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados"(Samuel, 1959).

2.3.2 Breve Histórico do Aprendizado de Máquina

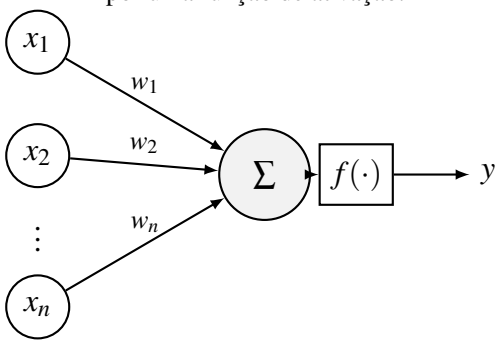
O desenvolvimento dos métodos modernos de aprendizado de máquina teve início com modelos inspirados na estrutura biológica do cérebro humano. Em 1943, McCulloch e Pitts propuseram o primeiro neurônio artificial, baseado em uma soma ponderada seguida de um limiar fixo (McCulloch; Pitts, 1943). Embora inovador, esse modelo não possuía mecanismo de ajuste ou aprendizado.

A limitação foi superada em 1958, quando Frank Rosenblatt introduziu o Perceptron, o primeiro algoritmo capaz de modificar seus pesos com base em exemplos rotulados (Rosenblatt, 1958). O Perceptron constitui o ancestral direto das redes neurais modernas, estabelecendo o princípio fundamental de aprendizado supervisionado baseado em correção de erro. A Figura 11 ilustra a estrutura básica do Perceptron proposto por Rosenblatt.

2.4 Aprendizado de Máquina

O Aprendizado de Máquina (*Machine Learning* (ML)) é um subcampo da Inteligência Artificial que reúne técnicas capazes de identificar padrões em dados e realizar previsões ou

Figura 11 – Estrutura do Perceptron de Rosenblatt: cada entrada é ponderada por um peso ajustável e processada por uma função de ativação.



Fonte: Elaboração do autor.

classificações automaticamente. Diferentemente de métodos baseados em regras estáticas, os modelos de ML aprendem relações a partir de exemplos históricos, ajustando seus parâmetros internos por meio de um processo de otimização estatística para minimizar erros de generalização (Murphy, 2012).

No contexto de sistemas fotovoltaicos, o ML tem se mostrado especialmente adequado devido ao elevado volume de dados gerados por inversores, *stringboxes* e estações meteorológicas, bem como pela natureza dinâmica e estocástica da geração solar. Modelos baseados em dados conseguem capturar variações complexas — como mudanças não-lineares de irradiância, comportamento térmico específico de cada unidade e padrões sutis de subperformance — que são extremamente difíceis de modelar explicitamente por equações determinísticas.

Segundo (Bishop, 2006), os algoritmos de aprendizado de máquina podem ser categorizados, com base na natureza do sinal de feedback disponível durante o treinamento, em três classes principais:

- **Aprendizado Supervisionado:** O modelo aprende uma função de mapeamento entre variáveis de entrada (X) e uma variável de saída (Y) a partir de um conjunto de dados rotulados (pares de entrada-saída conhecidos). É a abordagem utilizada quando se dispõe de um histórico de falhas classificadas por especialistas.
- **Aprendizado Não Supervisionado:** O algoritmo identifica estruturas, agrupamentos ou padrões ocultos nos dados sem a necessidade de rótulos prévios. É frequentemente aplicado para detecção de anomalias (*outlier detection*) em cenários onde não se sabe *a priori* o que constitui um comportamento normal ou falho.
- **Aprendizado Semissupervisionado:** Uma abordagem híbrida que combina uma pequena quantidade de dados rotulados com um grande volume de dados não rotulados. Esta técnica é particularmente valiosa no setor fotovoltaico, onde a aquisição de dados brutos é

barata e abundante, mas a rotulação de falhas é custosa e escassa.

2.4.1 Generalização, Overfitting e Divisão dos Dados

O objetivo fundamental de um modelo de aprendizado de máquina não é apenas reproduzir os dados históricos, mas sim adquirir a capacidade de **generalização**, ou seja, realizar predições precisas em dados novos e desconhecidos. Para garantir essa capacidade e avaliar o desempenho do modelo, adota-se uma metodologia rigorosa de divisão do *dataset* e monitoramento de métricas.

2.4.1.1 Overfitting e Underfitting

Durante o treinamento, o modelo busca encontrar uma função matemática que se ajuste aos dados de entrada. Nesse processo, dois fenômenos indesejados podem ocorrer, conforme ilustrado na **Figura 12**:

- ***Underfitting* (Subajuste):** Ocorre quando o modelo é muito simples para capturar a complexidade do fenômeno. No contexto fotovoltaico, seria como tentar prever a geração usando uma linha reta, ignorando a curvatura de sino da irradiância. O erro é alto tanto no treino quanto no teste.
- ***Overfitting* (Sobreajuste):** Ocorre quando o modelo é excessivamente complexo e começa a "decorar"o ruído estocástico dos dados de treinamento em vez de aprender o padrão subjacente. O modelo apresenta desempenho excelente no treino, mas falha drasticamente ao processar novos dados, pois aprendeu peculiaridades irrelevantes (como uma nuvem específica de um dia passado) em vez da física do sistema (Domingos, 2012).

2.4.1.2 Estratégia de Divisão dos Dados

Para evitar o *overfitting* e validar a eficácia do modelo, o conjunto total de dados é particionado em três subconjuntos distintos e excludentes:

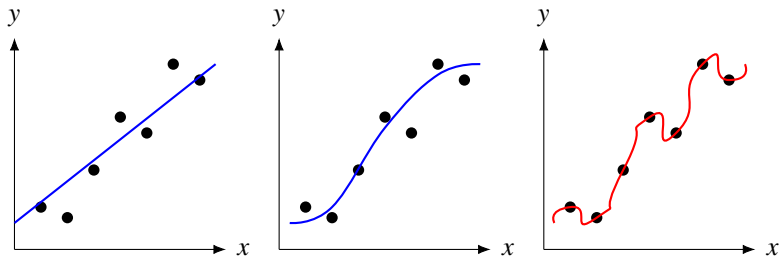
1. **Conjunto de Treinamento (Training Set):** A maior parcela dos dados (tipicamente 70-80%). É utilizado pelo algoritmo para ajustar seus parâmetros internos (pesos das árvores, no caso do XGBoost).
2. **Conjunto de Validação (Validation Set):** Uma parcela intermediária (10-15%) utilizada durante o processo de treinamento para sintonia fina de hiperparâmetros e para a técnica

de *Early Stopping*. Se o erro na validação começar a subir enquanto o erro de treino cai, interrompe-se o treinamento, pois é um sinal claro de início de *overfitting* (Bishop, 2006).

3. **Conjunto de Teste (Test Set):** A parcela final (10-15%), mantida isolada até o fim do desenvolvimento. É usada apenas uma única vez para a avaliação final do modelo. Como esses dados nunca foram "vistos"pelo algoritmo durante o aprendizado ou ajuste, eles fornecem uma estimativa imparcial do desempenho real do sistema em operação.

No contexto deste trabalho, a divisão não é feita de forma aleatória simples, mas sim respeitando a integridade temporal ou por arquivo (dia de operação), garantindo que não haja vazamento de dados (*data leakage*) entre o treino e o teste.

Figura 12 – Ilustração dos conceitos de ajuste de modelo: (a) *Underfitting*, onde o modelo falha em capturar o padrão; (b) Ajuste Ideal; (c) *Overfitting*, onde o modelo incorpora o ruído dos dados, perdendo capacidade de generalização.



(a) *Underfitting* (b) Ajuste Ideal (c) Overfitting

Fonte: Elaboração do autor.

2.4.2 Métricas de Avaliação de Desempenho

A avaliação de modelos de classificação requer métricas que quantifiquem a capacidade do algoritmo em distinguir corretamente entre as classes de interesse. A base para o cálculo dessas métricas é a **Matriz de Confusão**, uma tabela de contingência que cruza as predições do modelo com os valores reais (rótulos) do conjunto de teste.

Conforme ilustrado na **Figura 13**, as predições são categorizadas em quatro grupos fundamentais:

- ***Verdadeiro Positivo* (TP):** O modelo detectou corretamente uma falha (ex: string parada identificada como tal).
- ***Verdadeiro Negativo* (TN):** O modelo identificou corretamente o funcionamento normal.
- ***Falso Positivo* (FP):** O modelo gerou um alarme falso, classificando uma operação normal como falha (Erro Tipo I).
- ***Falso Negativo* (FN):** O modelo falhou em detectar uma anomalia existente, classificando-a

como normal (Erro Tipo II).

A partir dessas contagens, derivam-se os seguintes indicadores de desempenho:

2.4.2.1 Acurácia (Accuracy)

É a métrica mais intuitiva, representando a proporção global de acertos do modelo.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

Contudo, em problemas de detecção de anomalias onde as classes são desbalanceadas (ex: 98% dos dados são normais), a acurácia pode ser enganosa. Um modelo que simplesmente prediz "Tudo Normal"teria 98% de acurácia, mas seria inútil para o propósito de diagnóstico.

2.4.2.2 Precisão (Precision)

Indica a confiabilidade dos alarmes gerados. Responde à pergunta: "Dentre todas as falhas que o modelo apontou, quantas eram reais?"

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{2.8}$$

No contexto operacional, baixa precisão implica em excesso de falsos alarmes, o que gera custos desnecessários de deslocamento das equipes de manutenção e descrédito do sistema de monitoramento ("fadiga de alarme").

2.4.2.3 Revocação (Recall ou Sensibilidade)

Mede a capacidade do modelo de encontrar as falhas existentes. Responde à pergunta: "Dentre todas as falhas que realmente aconteceram, quantas o modelo conseguiu capturar?"

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.9}$$

Para a detecção de anomalias fotovoltaicas, o *Recall* é frequentemente priorizado, pois um Falso Negativo (deixar uma string queimada operando sem reparo) resulta em perda financeira direta e irreversível de geração (Japkowicz; Shah, 2011).

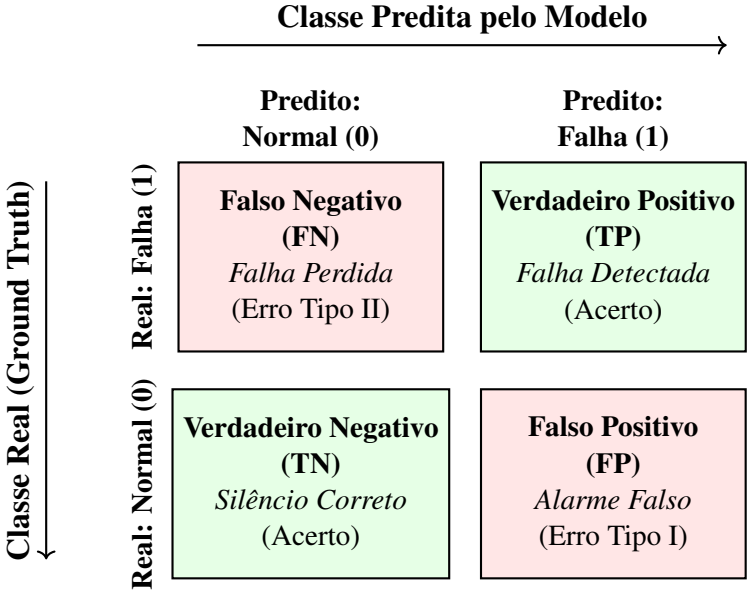
2.4.2.4 F1-Score

O F1-Score é a média harmônica entre Precisão e Recall, fornecendo uma métrica única que penaliza modelos que priorizam excessivamente uma métrica em detrimento da outra.

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{2.10}$$

Esta é a métrica principal utilizada para comparação de modelos neste trabalho, pois busca um equilíbrio entre evitar falsos alarmes (Precisão) e garantir a detecção das anomalias (Recall).

Figura 13 – Estrutura da Matriz de Confusão (Normal vs. Falha)



Fonte: Elaboração do autor.

2.4.3 Validação Cruzada (Cross-Validation)

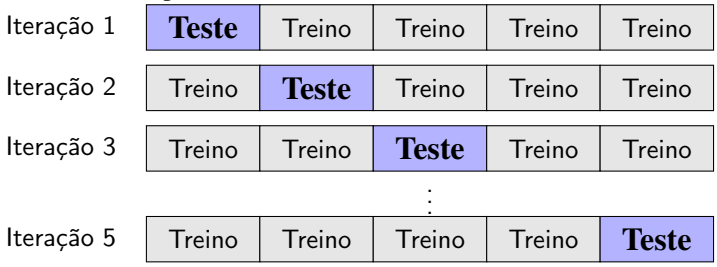
Em cenários onde o conjunto de dados é limitado ou apresenta desbalanceamento severo de classes, a divisão estática dos dados em conjuntos de treino e teste (método *Hold-out*) pode introduzir vieses significativos na avaliação do modelo. Dependendo de como os dados são sorteados, o conjunto de teste pode acabar contendo amostras "fáceis"(superestimando o desempenho) ou "difíceis"(subestimando-o).

Para mitigar esse problema, utiliza-se a técnica de Validação Cruzada (*Cross-Validation*). A variante mais comum, o *K-Fold*, consiste em particionar o conjunto de dados em k subconjuntos (ou "dobras") de tamanho aproximadamente igual. O processo de treinamento e avaliação é repetido k vezes, conforme ilustrado na Figura 14.

Em cada iteração i, a i-ésima dobra é reservada para teste (validação), enquanto as $k-1$ dobras restantes são utilizadas para treinamento. O desempenho final do modelo é reportado como a média das métricas obtidas nas k iterações, frequentemente acompanhada do desvio padrão, fornecendo uma estimativa estatisticamente mais robusta da capacidade de generalização do algoritmo (Bishop, 2006).

Quando se trata de classificação, utiliza-se o ***Stratified K-Fold***, que garante que a proporção original das classes seja mantida em cada dobra, assegurando que o modelo seja sempre treinado e testado com exemplos de todas as categorias.

Figura 14 – Esquema da Validação Cruzada K-Fold com $k = 5$. Em cada iteração, uma parte diferente dos dados (azul) é usada para validar o modelo treinado no restante (cinza).



Fonte: Elaboração do autor.

2.5 Algoritmos Relevantes para Detecção de Anomalias

A detecção de anomalias em sistemas fotovoltaicos tem sido abordada na literatura por uma ampla diversidade de algoritmos de aprendizado de máquina. O espectro de soluções abrange desde métodos clássicos de classificação, como **Support Vector Machines (SVM)** e **k-Nearest Neighbors (kNN)**, até abordagens baseadas em aprendizado profundo (*Deep Learning*), como as **Redes Neurais Artificiais**.

Paralelamente, em cenários onde a rotulação de dados é escassa, destacam-se métodos não supervisionados baseados em densidade e isolamento, como **Local Outlier Factor (LOF)** e **Isolation Forest**. Cada técnica apresenta vantagens e limitações específicas em função da não-linearidade da geração solar, do desbalanceamento entre classes (falhas são eventos raros) e do custo computacional de treinamento.

Mais recentemente, algoritmos de *ensemble* baseados em árvores de decisão, notadamente o **XGBoost**, emergiram como o estado da arte para dados tabulares industriais. Sua capacidade de modelar fronteiras de decisão complexas, aliada à robustez frente a ruídos e dados faltantes, o torna particularmente aderente aos desafios do monitoramento de *stringboxes*.

A seguir, são discutidos os princípios fundamentais dos principais algoritmos utiliza-

dos no setor, contextualizando suas aplicações e justificando a escolha do método proposto neste trabalho.

2.5.1 Regressão Linear

A Regressão Linear é um dos métodos mais fundamentais e amplamente utilizados na estatística e no aprendizado de máquina supervisionado. Seu objetivo principal é modelar a relação entre uma variável dependente (alvo) e um ou mais vetores de variáveis independentes (atributos), assumindo que essa relação pode ser descrita por uma função linear.

Em sua forma mais simples (regressão linear simples), o modelo busca ajustar uma reta que melhor descreve a distribuição dos dados. Generalizando para múltiplas variáveis, a predição $\hat{y}$ é dada pela soma ponderada dos atributos de entrada:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon \tag{2.11}$$

Onde:

- $x_1, \ldots, x_n$: São as características (*features*) do objeto de estudo;
- β_0: É o intercepto (viés), que indica o valor esperado quando todas as entradas são nulas;
- $\beta_1, \ldots, \beta_n$: São os coeficientes angulares que determinam o peso e a influência de cada variável na saída;
- ε: Representa o termo de erro ou resíduo aleatório não explicado pelo modelo.

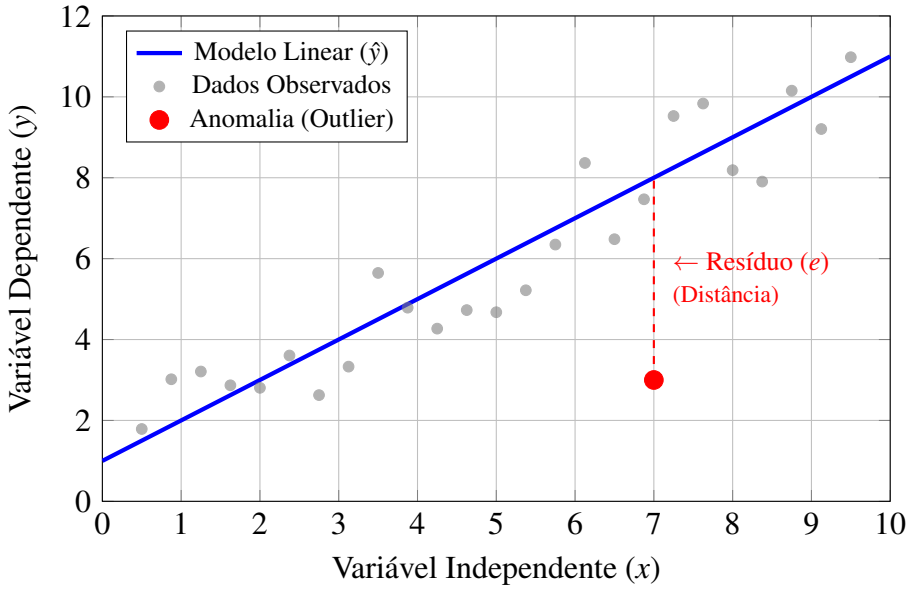
O ajuste dos parâmetros (β) é tipicamente realizado através do Método dos Mínimos Quadrados Ordinários (OLS), que busca minimizar a soma dos quadrados das diferenças entre os valores observados e os valores preditos pelo modelo linear.

2.5.1.1 Aplicação em Detecção de Anomalias

Embora seja um algoritmo de regressão, métodos lineares são frequentemente adaptados para a detecção de anomalias através da análise de resíduos. O princípio baseia-se na premissa de que o modelo aprende o "comportamento médio"ou normal dos dados.

Dado o resíduo $e = y_{real} - \hat{y}$, define-se um limiar de aceitação (geralmente baseado no desvio padrão σ da distribuição dos erros). Pontos de dados cuja distância em relação à reta de regressão excede esse limiar ($|e| > k\sigma$) são estatisticamente improváveis segundo o modelo linear, sendo, portanto, classificados como anomalias ou *outliers*.

Figura 15 – Representação visual da Regressão Linear: a reta azul minimiza o erro médio dos dados normais (cinza). A anomalia (vermelho) é identificada pelo alto valor residual (distância vertical em relação à reta).



Fonte: Elaboração do autor.

2.5.2 Modelos Baseados em Árvores de Decisão

Dentre as diversas abordagens de aprendizado de máquina para dados estruturados (tabulares), os modelos baseados em árvores destacam-se pela sua intuitividade e capacidade de modelar fronteiras de decisão não-lineares.

2.5.2.1 Árvores de Decisão

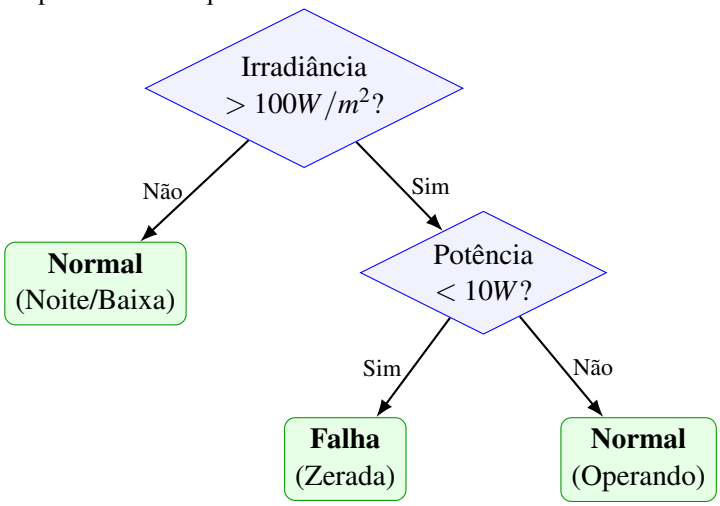
As Árvores de Decisão são modelos hierárquicos que particionam o espaço de dados em regiões retangulares distintas através de uma sequência de regras de decisão binárias ("se-então"). A estrutura, ilustrada na **Figura 16**, assemelha-se a um fluxograma, composto por:

- **Nó Raiz:** Contém todo o conjunto de dados inicial.
- **Nós de Decisão:** Pontos onde uma variável (ex: Irradiância) é testada contra um limiar (ex: $< 200 W/m^2$), dividindo os dados em subconjuntos.
- **Folhas:** Nós terminais que atribuem a classificação final (ex: "Normal"ou "Falha").

Segundo (Breiman *et al.*, 1984), a principal vantagem deste método é a interpretabilidade: o caminho percorrido da raiz até a folha revela a lógica exata utilizada para classificar uma anomalia. No entanto, árvores individuais sofrem de alta variância e instabilidade. Pequenas alterações nos dados de treinamento (como o ruído natural de sensores fotovoltaicos) podem gerar estruturas de árvore completamente diferentes, levando ao *overfitting* e baixa capacidade

de generalização.

Figura 16 – Estrutura de uma Árvore de Decisão simplificada para detecção de falhas: o espaço de dados é particionado sequencialmente com base em atributos físicos.



Fonte: Elaboração do autor.

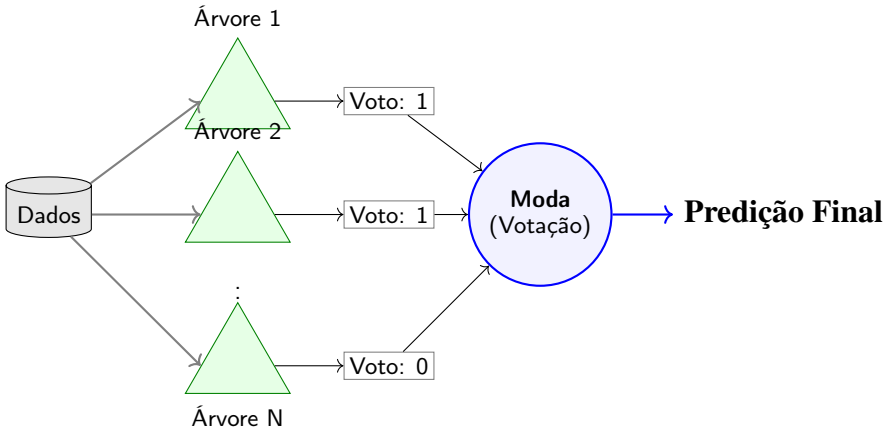
2.5.2.2 Random Forest (Floresta Aleatória)

Para mitigar a instabilidade das árvores individuais, (Breiman, 2001) propôs o algoritmo *Random Forest*. Esta técnica utiliza o conceito de *Bagging* (*Bootstrap Aggregating*) para criar um *Ensemble* (conjunto) formado por centenas de árvores de decisão treinadas em paralelo.

A robustez do método baseia-se em dois princípios de aleatoriedade. O primeiro consiste no treinamento de cada árvore a partir de uma subamostra aleatória dos dados, obtida com reposição. O segundo está relacionado ao processo de divisão dos nós, no qual apenas um subconjunto aleatório de *features* (variáveis) é considerado a cada divisão.

O resultado final é obtido pela votação majoritária (moda) das classes preditas por todas as árvores, conforme a Figura 17. Essa "sabedoria das multidões"reduz drasticamente a variância do modelo, tornando-o muito mais resistente a ruídos e *outliers* do que uma árvore única. Contudo, em cenários de dados desbalanceados (poucas falhas), o *Random Forest* pode ter dificuldade em capturar padrões muito sutis, uma limitação que outros métodos buscam resolver focando especificamente nos erros de classificação.

Figura 17 – Conceito de *Random Forest*: múltiplas árvores independentes votam para decidir a classe final, aumentando a robustez contra ruídos.



Fonte: Elaboração do autor.

2.5.3 Métodos de Margem e Distância

Diferentemente das abordagens baseadas em árvores, que particionam o espaço de forma hierárquica, esta classe de algoritmos fundamenta-se na geometria dos dados, utilizando conceitos de distância euclidiana e maximização de margens para estabelecer as fronteiras de decisão.

2.5.3.1 Support Vector Machines (SVM)

O *Support Vector Machine* (SVM) é um algoritmo de aprendizado supervisionado que busca encontrar o hiperplano ótimo que separa as classes de dados com a maior margem possível. Conforme ilustrado na Figura 18(a), os pontos de dados mais próximos da fronteira de decisão são chamados de "vetores de suporte"e são os únicos que determinam a posição do hiperplano (Bishop, 2006).

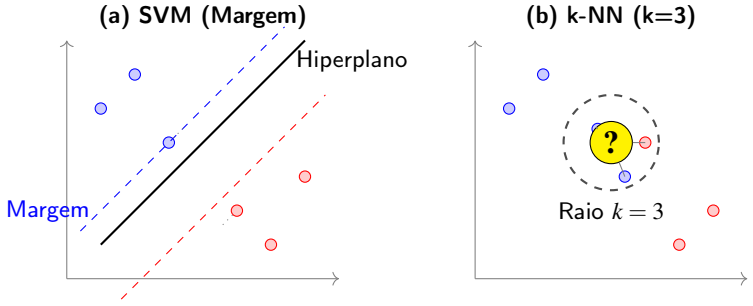
A grande vantagem do SVM reside no uso de funções de *Kernel* (como o RBF - *Radial Basis Function*), que permitem mapear dados não-lineares para dimensões superiores onde se tornam linearmente separáveis. No entanto, para o monitoramento de usinas fotovoltaicas, o SVM apresenta uma limitação crítica de escalabilidade. Sua complexidade computacional de treinamento cresce cubicamente com o número de amostras ($O(n^3)$). Em cenários com múltiplas *stringboxes* e histórico de longa duração, o custo computacional torna-se proibitivo para re-treinamentos frequentes.

2.5.3.2 k-Nearest Neighbors (k-NN)

O algoritmo k-NN (*k-Vizinhos Mais Próximos*) é um método não-paramétrico baseado na premissa de que dados similares ocupam regiões próximas no espaço de características. Para classificar uma nova amostra (ex: uma leitura de potência), o algoritmo calcula a distância (usualmente Euclidiana) em relação a todos os pontos do conjunto de treinamento e atribui a classe majoritária entre os k vizinhos mais próximos, como mostrado na Figura 18(b).

Apesar de sua simplicidade conceitual e intuitiva, o k-NN apresenta desvantagens severas para aplicação em tempo real. Por se tratar de um método de *aprendizado preguiçoso* (*lazy learner*), todo o processamento ocorre na etapa de predição, exigindo o cálculo de distâncias em relação a todo o histórico de dados a cada nova leitura. Além disso, o algoritmo é altamente sensível a ruídos e *outliers*, o que é particularmente crítico em usinas solares, onde sensores podem apresentar flutuações momentâneas, levando à classificação incorreta de uma operação normal caso existam vizinhos ruidosos próximos (Murphy, 2012). Por fim, o k-NN apresenta forte dependência da escala das variáveis, tornando indispensável uma normalização rigorosa dos dados, uma vez que variáveis com escalas maiores, como a irradiância no intervalo de 0 a 1000, tendem a dominar o cálculo da distância em detrimento de variáveis de menor escala, como a tensão normalizada entre 0 e 1.

Figura 18 – Comparação geométrica: (a) SVM busca o hiperplano (linha sólida) que maximiza a margem entre as classes; (b) k-NN classifica um novo ponto (interrogação) baseando-se na maioria dos vizinhos dentro de um raio de distância.



Fonte: Elaboração do autor.

2.5.4 Modelos não supervisionados baseados em densidade e isolamento

Em cenários onde a rotulação de dados históricos é escassa ou inexistente — uma situação comum em usinas recém-comissionadas —, a detecção de anomalias depende de abordagens não supervisionadas. Diferentemente dos classificadores apresentados anteriormente, estes algoritmos não buscam mapear uma entrada para uma classe conhecida, mas sim identificar

instâncias que divergem estatisticamente do padrão predominante dos dados. Duas técnicas destacam-se nesta categoria: o *Isolation Forest* e o *Local Outlier Factor* (LOF).

2.5.4.1 Isolation Forest (iForest)

O *Isolation Forest*, proposto por (Liu *et al.*, 2008), parte de uma premissa distinta da maioria dos algoritmos de detecção: em vez de modelar o comportamento normal para identificar desvios, ele busca isolar explicitamente as anomalias. O método fundamenta-se em duas propriedades das anomalias: elas são **minoritárias** (poucas instâncias) e possuem valores de atributos **distintos** daqueles considerados normais.

O algoritmo constrói um *ensemble* de árvores aleatórias (similar ao conceito de *Random Forest*), onde cada nó seleciona aleatoriamente uma característica e um valor de corte. A lógica, ilustrada na Figura 19(a), é que pontos anômalos são “mais fáceis” de isolar, exigindo menos cortes (divisões) para ficarem sozinhos em um nó terminal. Consequentemente, anomalias tendem a ter caminhos mais curtos da raiz até a folha da árvore, enquanto pontos normais, por estarem em regiões densas, exigem muitas divisões para serem isolados.

Para o monitoramento de *stringboxes*, o iForest é computacionalmente eficiente ($O(n)$), mas pode apresentar dificuldades em distinguir tipos específicos de falhas que possuem densidades similares, agrupando-as apenas como "anomalias genéricas".

2.5.4.2 Local Outlier Factor (LOF)

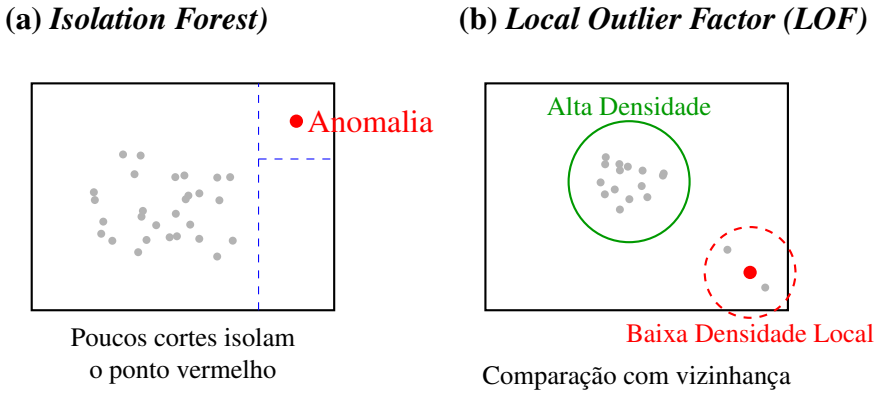
Enquanto o iForest baseia-se no isolamento global, o *Local Outlier Factor* (LOF), introduzido por (Breunig *et al.*, 2000), foca na densidade local dos dados. O algoritmo atribui a cada ponto um grau de anomalia baseado na razão entre sua densidade local e a densidade média de seus k-vizinhos mais próximos.

Conforme representado na Figura 19(b), regiões de operação normal (como a curva típica de geração em dias claros) apresentam alta densidade de pontos. Uma anomalia, como uma *stringbox* operando com potência reduzida em um horário de pico, aparecerá em uma região de baixa densidade (distante dos vizinhos). Se o LOF de um ponto é significativamente maior que 1, indica que ele está em uma região mais esparsa que seus vizinhos, caracterizando-o como um *outlier*.

A principal limitação do LOF para aplicações em tempo real é sua complexidade computacional ($O(n^2)$), que exige o cálculo de distâncias par-a-par, tornando-o lento para grandes

históricos de dados de usinas solares (Mellit; Kalogirou, 2008).

Figura 19 – Comparativo conceitual de detecção não supervisionada.



Fonte: Elaboração do autor.

Ambos os métodos são eficazes para detectar que "algo está errado". No entanto, para o objetivo deste trabalho — que consiste não apenas em detectar, mas em **diagnosticar** o tipo de falha (se a *stringbox* está zerada, sombreada ou com subperformance) — abordagens supervisionadas como o XGBoost levam vantagem, pois aprendem as assinaturas específicas de cada classe de falha a partir dos dados rotulados.

2.5.5 Redes Neurais Artificiais: Multilayer Perceptron (MLP)

As Redes Neurais Artificiais (RNA) são modelos computacionais inspirados na estrutura biológica do sistema nervoso, projetados para reconhecer padrões complexos através do aprendizado a partir de exemplos. Enquanto o Perceptron simples (discutido na Seção 2.3.2) é limitado a problemas linearmente separáveis, o *Multilayer Perceptron* (MLP) supera essa restrição através da introdução de camadas intermediárias de neurônios, conhecidas como camadas ocultas (*hidden layers*).

2.5.5.1 Arquitetura e Funcionamento

O MLP é uma rede do tipo *feedforward* (alimentação direta), composta por pelo menos três camadas de nós: uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Conforme ilustrado na Figura 20, cada neurônio em uma camada conecta-se a todos os neurônios da camada subsequente através de conexões ponderadas (pesos sinápticos).

Matematicamente, o processamento em um neurônio j da camada oculta pode ser descrito pela soma ponderada das entradas seguida por uma função de ativação não-linear $\phi(\cdot)$:

$$y_j = \phi\left(\sum_{i=1}^{n} w_{ji} x_i + b_j\right) \tag{2.12}$$

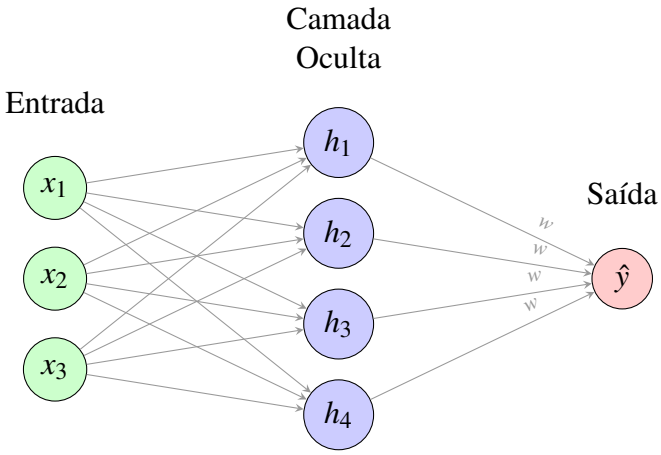
Onde w_{ji} representa o peso da conexão entre a entrada i e o neurônio j, e b_j é o viés (*bias*). A presença da função de ativação não-linear (como a Sigmoide, Tangente Hiperbólica ou ReLU) é fundamental: segundo o Teorema da Aproximação Universal (Hornik *et al.*, 1989), um MLP com apenas uma camada oculta e neurônios suficientes é capaz de aproximar qualquer função contínua com precisão arbitrária.

2.5.5.2 O Algoritmo de Backpropagation

O treinamento do MLP ocorre de forma supervisionada por meio do algoritmo de *Backpropagation* (retropropagação do erro), popularizado por (Rumelhart *et al.*, 1986). Esse processo é composto por duas fases principais. Na fase de propagação direta (*forward*), os dados fluem da camada de entrada até a camada de saída, resultando na geração de uma predição. Em seguida, na fase de retropropagação (*backward*), o erro associado à predição é calculado e propagado no sentido inverso, da saída para a entrada. Nesse estágio, utilizando o método do gradiente descendente, os pesos w da rede são ajustados iterativamente com o objetivo de minimizar a função de custo global.

No contexto de sistemas fotovoltaicos, MLPs são frequentemente utilizados para modelar a relação não-linear entre irradiância, temperatura e potência (Mellit; Kalogirou, 2008). Contudo, apresentam desvantagens como a natureza de "caixa-preta"(baixa interpretabilidade), a necessidade de grandes volumes de dados para treinamento e a sensibilidade ao ajuste de hiperparâmetros (número de camadas, neurônios e taxa de aprendizado).

Figura 20 – Arquitetura de um Multilayer Perceptron (MLP) com uma camada oculta.



A não-linearidade é introduzida
na camada oculta.

Fonte: Elaboração do autor.

2.5.6 O Algoritmo XGBoost

Neste trabalho, adota-se uma abordagem supervisionada baseada no algoritmo XGBoost. Proposto originalmente por (Chen; Guestrin, 2016), este método pertence à família dos algoritmos de *Ensemble* baseados em árvores de decisão e destaca-se por sua alta capacidade de generalização, eficiência computacional em grandes bases de dados e robustez no tratamento de valores ausentes.

A principal distinção do XGBoost em relação a outros métodos de *ensemble*, como o *Random Forest*, reside na sua estratégia de construção. Enquanto o *Random Forest* treina árvores independentes em paralelo (*Bagging*), o XGBoost utiliza a técnica de **Boosting** (aprendizado sequencial). Nesta abordagem, os modelos não são criados para prever a variável alvo diretamente, mas sim para corrigir os erros residuais dos modelos anteriores.

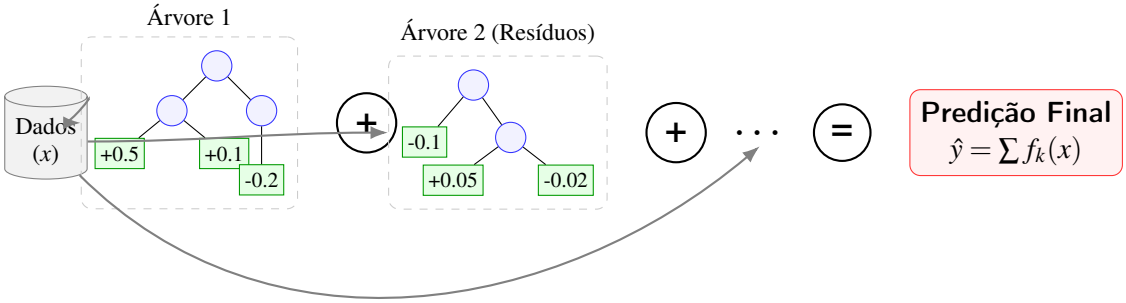
A lógica funciona da seguinte maneira, conforme ilustrado na Figura 21. Inicialmente, o algoritmo constrói uma primeira árvore simples responsável por gerar uma previsão inicial. Em seguida, é calculado o erro, ou resíduo, dessa previsão em relação ao valor real observado. A árvore subsequente não busca estimar diretamente o alvo original, mas sim modelar o erro cometido pela árvore anterior. Esse procedimento ocorre de forma sequencial, de modo que cada nova árvore adicionada ao modelo contribui para o refinamento da predição, corrigindo progressivamente as falhas das árvores precedentes.

Matematicamente, a predição final ($\hat{y}_i$) para um dado de entrada não vem de uma única árvore, mas da soma dos resultados de todas as K árvores criadas:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{2.13}$$

Onde f_k representa a predição da k-ésima árvore. Além de corrigir erros, o XGBoost possui um mecanismo interno chamado Regularização (Ω), que impede que as árvores fiquem complexas demais. Isso é fundamental para evitar que o modelo "decore"os dados de treinamento (*overfitting*) e garante que ele funcione bem em dados novos, como na detecção de falhas reais em dias futuros.

Figura 21 – Princípio de funcionamento do XGBoost (Boosting): as árvores são adicionadas sequencialmente, onde cada novo modelo corrige o erro residual do anterior.



Fonte: Elaboração do autor.

2.6 Trabalhos Relacionados

A aplicação de técnicas de inteligência artificial para o monitoramento de sistemas fotovoltaicos tem sido amplamente explorada na literatura recente. As abordagens variam conforme a granularidade dos dados (módulo, *string* ou inversor) e a complexidade dos algoritmos utilizados.

Uma vertente significativa da literatura foca no uso de *Aprendizado Profundo (Deep Learning)*. (Chen *et al.*, 2019) propuseram um método baseado em Redes Neurais Convolucionais (CNN) treinadas com mapas de características extraídos de curvas I-V para classificar falhas como curto-circuito e sombreamento parcial. Embora alcancem alta precisão, tais métodos exigem equipamentos dedicados para traçagem de curvas I-V ou câmeras térmicas, o que inviabiliza a aplicação em larga escala em usinas que contam apenas com o sistema SCADA padrão.

No domínio dos métodos baseados em dados convencionais, (Dhimish *et al.*, 2018) desenvolveram algoritmos baseados em análise estatística e lógica t-test para detectar falhas em nível de string. O método compara a potência teórica com a medida, identificando anomalias

quando o desvio supera um limiar. Entretanto, a dependência de limiares estáticos definidos manualmente torna o sistema rígido, dificultando a adaptação para diferentes usinas ou condições climáticas dinâmicas sem recalibração constante.

O uso de algoritmos de *ensemble*, como Random Forest e XGBoost, também tem crescido. (Pierro *et al.*, 2019) aplicaram métodos de *Gradient Boosting* focados na previsão de geração de energia (forecasting) para detecção de desvios. Contudo, a maioria desses estudos realiza a análise no nível do inversor central, o que mascara falhas localizadas (como uma única stringbox com fusível queimado) devido ao efeito de média da alta potência agregada.

Diferentemente das abordagens citadas, este trabalho propõe uma metodologia focada no nível intermediário (*stringbox*), utilizando o algoritmo XGBoost não apenas para regressão, mas para classificação multiclasse via estratégia *One-vs-All*. A principal inovação reside na engenharia de atributos baseada na morfologia da curva (como *skewness* e a referência dinâmica Q85), permitindo a distinção entre sombreamento e subperformance sem a necessidade de sensores meteorológicos de alta precisão ou imagens externas.

3 METODOLOGIA

Este capítulo descreve os procedimentos metodológicos adotados para o desenvolvimento do sistema de detecção de anomalias. A abordagem é dividida em cinco etapas principais: caracterização da base de dados, pré-processamento e engenharia de atributos, definição dos critérios de rotulagem (*ground truth*), configuração do algoritmo XGBoost e, por fim, as métricas utilizadas para avaliação de desempenho.

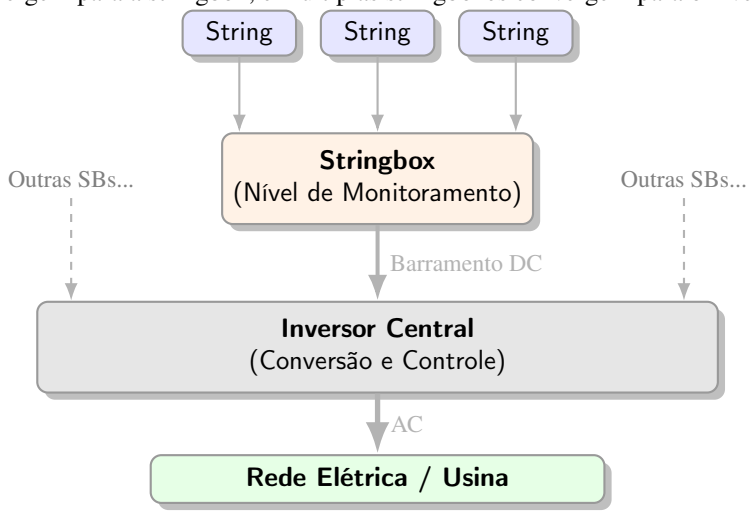
3.1 Objeto de Estudo

O presente trabalho utiliza dados operacionais reais provenientes de uma usina fotovoltaica de grande porte (*utility-scale*) localizada no Nordeste do Brasil. Cujo dados foram anonimizados para preservar a confidencialidade do empreendimento.

A usina analisada possui capacidade instalada superior a 200 MWp e adota uma arquitetura baseada em inversores centrais, nos quais múltiplas *stringboxes* inteligentes realizam o agrupamento e o monitoramento das séries fotovoltaicas. Cada *stringbox* fornece medições contínuas de grandezas elétricas (tensão, corrente e potência agregada), que compõem a base de dados utilizada para o desenvolvimento e validação dos métodos de detecção de anomalias propostos neste estudo.

A Figura 22 ilustra de forma simplificada a hierarquia elétrica típica de uma usina fotovoltaica, destacando o posicionamento da stringbox no sistema.

Figura 22 – Representação da hierarquia de agregação em uma usina com topologia central: múltiplas strings convergem para a stringbox, e múltiplas stringboxes convergem para o inversor.



Fonte: Elaboração do autor.

3.2 Premissas Operacionais e Delimitações do Estudo

Para garantir clareza metodológica e correta interpretação dos resultados apresentados, esta seção explicita as premissas operacionais adotadas no desenvolvimento do sistema, bem como as delimitações de escopo assumidas ao longo do trabalho.

A usina fotovoltaica analisada opera com estrutura de seguimento solar em um eixo (*single-axis tracking*). No entanto, embora o comportamento dinâmico da geração reflita a presença de rastreadores solares, falhas específicas de trackers não são tratadas explicitamente neste estudo. Assim, anomalias associadas a desalinhamento mecânico, travamento ou falha de atuadores de rastreamento não constituem uma classe de falha dedicada, podendo eventualmente manifestar-se como padrões de subperformance ou distorções temporais não classificadas.

Outra premissa relevante refere-se à disponibilidade de dados. Stringboxes que apresentem ausência total de dados de potência ao longo do dia — seja por falha elétrica, desligamento operacional ou perda de comunicação — são classificadas como casos de "Zerada". Do ponto de vista do sistema de monitoramento, a indisponibilidade completa de medições representa uma condição operacional crítica equivalente à perda total de geração daquele equipamento, justificando sua inclusão nesta classe de anomalia.

Essa decisão metodológica reflete uma abordagem prática alinhada ao contexto industrial, na qual o diagnóstico inicial prioriza a identificação de ativos indisponíveis, independentemente da causa raiz específica (elétrica ou comunicacional), que pode ser posteriormente investigada pela equipe de manutenção.

Por fim, destaca-se que o sistema proposto tem como foco a detecção automática de anomalias no nível de *stringboxes*, utilizando exclusivamente dados elétricos e irradiância, sem integração com sensores mecânicos ou estados internos de trackers. Essa delimitação permite avaliar a eficácia de métodos baseados em aprendizado de máquina mesmo em cenários com instrumentação limitada.

3.3 Base de Dados e Seleção de Amostras

A construção do *dataset* adotou uma estratégia de amostragem baseada em eventos (*Event-Based Sampling*). Em vez de processar períodos cronológicos contínuos de toda a usina — o que resultaria em um severo desbalanceamento de classes devido à raridade natural das falhas —, foram selecionados diversos arquivos de dados diários provenientes de múltiplos inversores

centrais.

Essa seleção priorizou dias nos quais anomalias foram oficialmente registradas pela equipe de operação (sombreamento e subperformance), além de dias representativos de operação normal. Essa combinação garante diversidade operacional e permite que o modelo aprenda a distinguir padrões entre diferentes condições ambientais e equipamentos.

3.4 Extração de Características (*Feature Engineering*)

Diferentemente de abordagens que inserem a série temporal bruta diretamente no classificador, este trabalho adotou uma estratégia de extração de características estatísticas e temporais. O objetivo é sintetizar o comportamento diário de cada *stringbox* em um vetor de atributos fixo, capaz de capturar a "assinatura"morfológica de cada tipo de falha.

O algoritmo desenvolvido, denominado *Feature Extractor*, processa os arquivos JSON brutos e calcula 28 indicadores estatísticos para cada equipamento. Esses atributos, detalhados na Tabela 2, foram selecionados para capturar tanto a magnitude da perda de potência quanto a forma da curva de geração.

Tabela 2 – Conjunto de características extraídas das séries temporais para treinamento dos modelos XGBoost.

Categoria	**Atributos Extraídos**	**Significado Físico/Estatístico**
Estatística Básica	Média, Mediana, Desvio Padrão	Tendência central e dispersão da potência diária.
	Mínimo, Máximo, Amplitude	Faixa dinâmica de operação.
	Percentis (25%, 75%) e IQR	Distribuição robusta a *outliers*.
	Coeficiente de Variação (CV)	Instabilidade da geração (σ/μ).
	Zeros (Contagem e Razão)	Indicador primário para falha total ou perda de comunicação.
Forma da Distribuição	Assimetria (*Skewness*)	Identifica caudas na distribuição (típico de sombreamento parcial).
	Curtose (*Kurtosis*)	Identifica achatamento da curva (típico de limitação de potência).
Desvio da Referência	Média do Desvio (Δ_{ref})	Diferença média $P_{SB} - P_{ref}$ (viés de performance).
	Desvio Padrão do Desvio	Estabilidade do erro em relação à referência Q85.
	Correlação $P_{SB} \times G_{POA}$	Linearidade da resposta à irradiância.
Assinaturas Temporais	Deltas por Período	Desvio médio na Manhã (8-12h), Tarde (14-17h) e Noite.
	Hora de Melhor/Pior Delta	Momento do dia com maior/menor desvio relativo.
	Razão Manhã/Tarde	Comparativo de produção entre períodos (detecção de obstáculo fixo).
	Variabilidade Temporal	Oscilação do desvio de performance ao longo do dia.
	Consistência Temporal	Razão entre estabilidade e magnitude do desvio.
Contexto	Irradiância Média/Máxima	Disponibilidade do recurso solar no dia analisado.

Fonte: Elaboração do autor.

3.4.1 Exemplificação da Estrutura de Dados

A fim de ilustrar a organização dos dados após o processamento dos arquivos JSON originais, a Tabela 3 apresenta um recorte amostral cobrindo três momentos distintos do dia (manhã, meio-dia e tarde).

Este formato tabular permite observar a magnitude numérica das falhas:

– **SB01 (Normal):** Acompanha a referência (P_{ref}) em todos os horários com precisão.
– **SB05 (Subperformance):** Apresenta um *ratio* de perda constante (aprox. -20%) independente da irradiância ou horário.
– **SB06 (Sombreamento):** Apresenta desvio severo apenas no período matutino, recuperando sua performance nominal nos horários de sol a pino e vespertinos.

Tabela 3 – Exemplo de estrutura tabular dos dados processados, evidenciando os valores numéricos das anomalias.

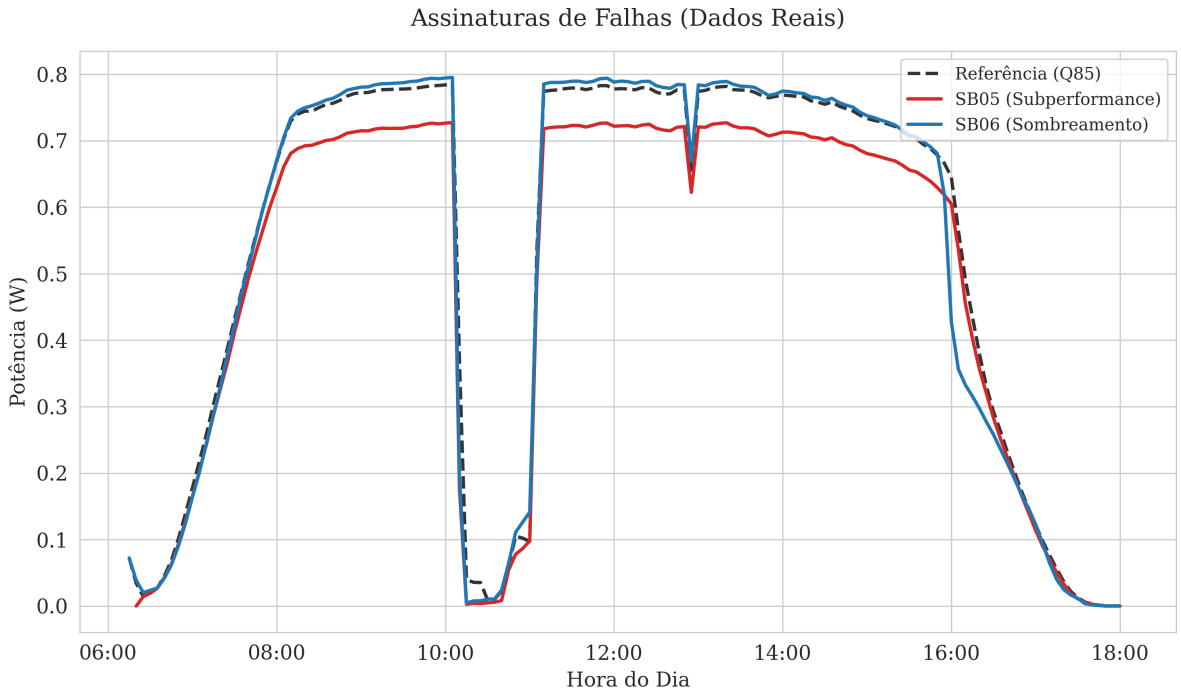
Timestamp *(Data/Hora)*	G_{POA} (W/m^2)	P_{ref} **(Q85)** (W)	P_{SB01} **(Normal)** (W)	P_{SB05} **(Subperf.)** (W)	P_{SB06} **(Sombre.)** (W)
Período Matutino (Sombreamento Ativo)					
2024-06-30 08:00:00	350.5	15200	15150	12160	7500
2024-06-30 08:05:00	365.2	15800	15780	12640	7800
2024-06-30 08:10:00	380.1	16500	16450	13200	8200
Período Meio-Dia (Alta Irradiância)					
2024-06-30 12:00:00	980.4	48000	47950	38400	47900
2024-06-30 12:05:00	985.1	48200	48150	38560	48100
2024-06-30 12:10:00	978.8	47900	47850	38320	47880
Período Vespertino (Sem Sombreamento)					
2024-06-30 16:00:00	410.5	18500	18450	14800	18400
2024-06-30 16:05:00	400.2	18000	17950	14400	17920
2024-06-30 16:10:00	390.8	17500	17480	14000	17450

Nota: Valores ilustrativos. Laranja: Perda constante (≈ 20%). Azul: Perda localizada.
Fonte: Elaboração do autor.

3.4.2 Análise Visual das Assinaturas de Falha

Complementando a visão tabular, realizou-se uma inspeção gráfica dos dados brutos utilizados para treinamento. A Figura 23 apresenta as curvas de potência (P_{SB}) do mesmo dia amostrado acima, representadas em relação à referência dinâmica (P_{ref}).

Figura 23 – Comparativo de curvas reais de potência evidenciando as assinaturas morfológicas distintas de Subperformance e Sombreamento.



Fonte: Elaboração do autor.

O gráfico confirma visualmente as premissas utilizadas na engenharia de atributos:

- **Subperformance (Linha Laranja):** A curva mantém a morfologia senoidal correta, mas apresenta um *offset* negativo constante (translação vertical) em relação à referência tracejada durante todo o dia.
- **Sombreamento (Linha Azul):** A curva sofre uma deformação severa ("mordida") no período matutino, descolando-se da referência, mas recupera o comportamento normal nos períodos de alta elevação solar.

Essa distinção clara no domínio do tempo justifica a extração de atributos de forma (como *Skewness* e Razão Manhã/Tarde) para alimentar os classificadores, uma vez que médias simples poderiam mascarar a natureza temporal do sombreamento.

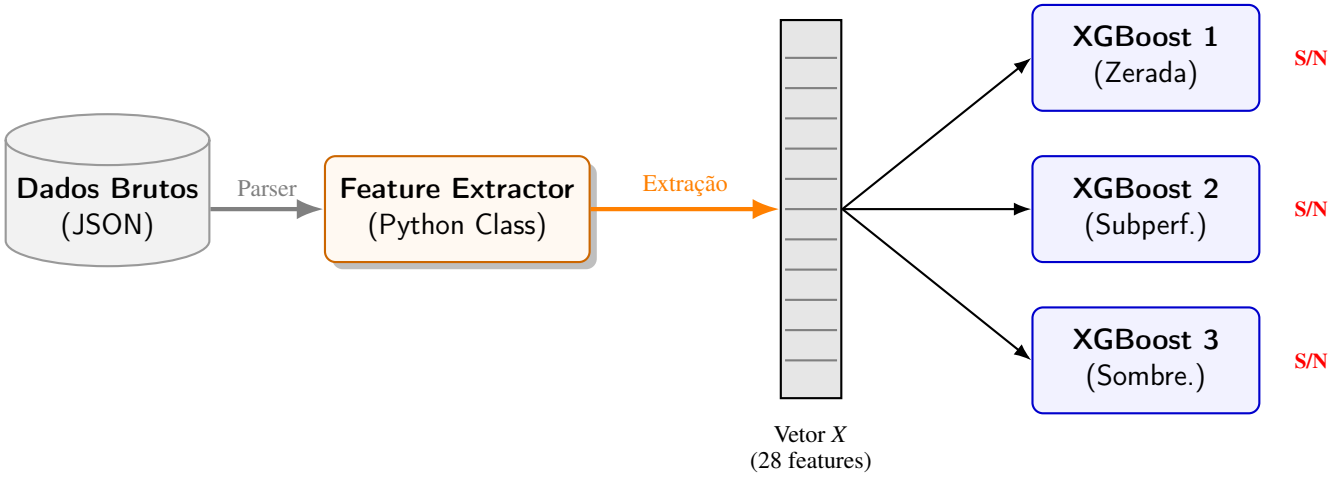
3.5 Arquitetura e Implementação

A implementação computacional foi realizada na linguagem Python, utilizando a biblioteca XGBoost. A arquitetura do sistema segue um fluxo de processamento em estágios, conforme ilustrado na Figura 24.

O processo inicia com a leitura dos arquivos brutos, passa pelo módulo de extração

de características (que condensa milhares de pontos em um vetor de 28 dimensões) e alimenta três classificadores independentes na estratégia *One-vs-All*.

Figura 24 – Fluxograma da arquitetura implementada: do dado bruto à decisão dos classificadores independentes.



Fonte: Elaboração do autor.

Esta arquitetura modular oferece vantagens significativas:

1. **Especialização:** Cada modelo aprende apenas os padrões estatísticos relevantes para sua falha específica (ex: o modelo de Sombreamento foca na *skewness*, enquanto o de Zerada foca na contagem de zeros).
2. **Manutenibilidade:** Novos tipos de falhas podem ser adicionados treinando um novo classificador binário, sem necessidade de retreinar toda a rede.

3.6 Estratégia de Treinamento e Validação

Para garantir a confiabilidade dos resultados, a estratégia de validação foi adaptada conforme a disponibilidade de dados para cada classe de anomalia.

3.6.1 Divisão Hold-out (Zerada e Subperformance)

Para os classificadores de anomalias elétricas (Zerada e Subperformance), que possuem assinaturas mais estáveis e determinísticas, adotou-se o método de validação cruzada simples (*Hold-out*) com divisão por arquivos (contexto).

- **Treino (80% dos arquivos):** Utilizado para ajuste dos pesos.
- **Validação (20% dos arquivos):** Mantido isolado para teste final.

Essa abordagem preserva a integridade temporal dos dados, simulando o cenário real onde o

modelo treinado no passado deve operar em dias futuros.

3.6.2 Validação Cruzada Estratificada (Sombreamento)

Para o classificador de Sombreamento, identificou-se uma limitação na quantidade absoluta de amostras (*Small Data*), o que tornava a divisão simples por arquivos suscetível a vieses sazonais (ex: testar em dias onde a geometria solar não gerava sombras, resultando em métricas enganosas).

Para superar essa limitação, adotou-se a **validação cruzada estratificada com 5 dobras** (*Stratified 5-Fold Cross-Validation*). Nesse procedimento, o conjunto total de dados é particionado em cinco subconjuntos, preservando-se em cada um deles a proporção de casos de sombreamento, aproximadamente 5% do total. O modelo é então treinado e avaliado cinco vezes, alternando-se iterativamente o subconjunto utilizado como conjunto de teste, enquanto os demais são empregados para treinamento. O desempenho final reportado corresponde à média das métricas obtidas ao longo das cinco iterações. Essa técnica maximiza o aproveitamento dos dados disponíveis, assegurando que cada exemplo de sombreamento seja utilizado tanto no treinamento quanto na validação em momentos distintos, permitindo uma avaliação mais justa da capacidade do modelo em reconhecer o padrão morfológico associado à falha.

3.6.3 Balanceamento de Classes

Dada a raridade natural das falhas, os modelos utilizam o parâmetro *scale_pos_weight* calculado dinamicamente conforme a Equação 3.1:

$$\text{scale\_pos\_weight} = \frac{\text{Número de Negativos (Normal)}}{\text{Número de Positivos (Falha)}} \tag{3.1}$$

Isso força o algoritmo XGBoost a penalizar mais severamente os erros de classificação na classe minoritária, compensando o desbalanceamento do conjunto de dados sem a necessidade de descartar dados normais (*undersampling*).

3.6.4 Configuração dos Hiperparâmetros

Os modelos foram treinados com uma configuração fixa, definida empiricamente para garantir robustez e evitar *overfitting* em datasets de médio porte, conforme Tabela 4.

Tabela 4 – Hiperparâmetros utilizados nos classificadores XGBoost.

Hiperparâmetro	Valor	Função
`n_estimators`	100	Número de árvores de decisão (*boosting rounds*).
`max_depth`	5	Profundidade máxima (limita complexidade).
`learning_rate`	0.1	Taxa de aprendizado (η).
`subsample`	0.8	Amostragem de linhas por árvore (reduz variância).
`colsample_bytree`	0.8	Amostragem de colunas por árvore.
`scale_pos_weight`	Dinâmico	Ajuste de peso para classe de falha.

Fonte: Elaboração do autor.

3.7 Métricas de Avaliação

A avaliação de desempenho priorizou métricas que penalizam falsos negativos e falsos positivos, essenciais para sistemas de monitoramento industrial:

- **Precision (Precisão):** Avalia a confiabilidade dos alarmes gerados.
- **Recall (Sensibilidade):** Avalia a capacidade de não deixar passar falhas reais.
- **F1-Score:** Média harmônica entre Precisão e Recall, utilizada como indicador principal de qualidade.

3.8 Lógica de Decisão e Diagnóstico Múltiplo

Uma característica fundamental da arquitetura *One-vs-All* adotada é a independência estatística entre os classificadores. Diferentemente de uma classificação multiclasse tradicional (onde as classes são mutuamente excludentes, ou seja, a soma das probabilidades deve ser 100%), a abordagem proposta permite a detecção de anomalias simultâneas.

Na prática, cada um dos três modelos ($M_{zerada}, M_{sub}, M_{sombreamento}$) emite uma probabilidade independente $p \in [0,1]$. O diagnóstico final é composto pela agregação binária dessas saídas, considerando o limiar de decisão padrão de 0.5:

$$\text{Diagnóstico} = \begin{cases} \text{Zerada}, & \text{se } p_{zerada} > 0.5 \\ \text{Subperformance}, & \text{se } p_{sub} > 0.5 \\ \text{Sombreamento}, & \text{se } p_{sombreamento} > 0.5 \end{cases} \tag{3.2}$$

Essa flexibilidade permite que o sistema identifique cenários complexos, como uma *stringbox* que apresenta subperformance crônica (fusível queimado em uma entrada) e, simultaneamente, sofre sombreamento em horários específicos, gerando alertas para ambas as equipes de manutenção (elétrica e limpeza/poda).

3.9 Ambiente Computacional e Ferramentas

O desenvolvimento do sistema, desde o pré-processamento dos arquivos JSON até o treinamento e validação dos modelos, foi realizado utilizando a linguagem de programação **Python** (versão 3.10), escolhida por sua robustez em análise de dados e amplo ecossistema de bibliotecas de aprendizado de máquina.

As principais bibliotecas e *frameworks* utilizados estão listados no Quadro 5.

Tabela 5 – Ferramentas computacionais e bibliotecas utilizadas no desenvolvimento.

Ferramenta/Biblioteca	**Função no Projeto**
Python	Linguagem base para orquestração de todo o fluxo de dados.
Pandas / NumPy	Manipulação de estruturas tabulares e cálculos vetoriais.
XGBoost Library	Implementação otimizada do algoritmo *Gradient Boosting*.
Scikit-Learn	Funções de métricas (*F1-score*) e divisão de dados.
Matplotlib / Seaborn	Visualização de dados e geração de gráficos de análise.

Fonte: Elaboração do autor.

Todo o processamento foi executado em ambiente computacional local, validando a premissa de que a engenharia de atributos proposta (vetor de 28 características por dia) resulta em modelos leves e eficientes, não demandando hardware de alto desempenho (como GPUs) para sua operação.

4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados os resultados obtidos com a aplicação da metodologia proposta, bem como a análise crítica de seu desempenho frente às diferentes categorias de falhas. O objetivo não é apenas relatar números, mas interpretar sua relevância operacional, compreender o comportamento dos modelos em termos de generalização e avaliar se as decisões tomadas ao longo do desenvolvimento — como a engenharia de atributos, a escolha do algoritmo XGBoost e a estratégia *one-vs-all* — foram adequadas ao problema estudado.

A discussão resulta, portanto, da combinação entre três perspectivas complementares:

- avaliação estatística, por meio das métricas de classificação;
- coerência física, verificando se os modelos aprenderam padrões alinhados com o comportamento esperado de sistemas fotovoltaicos;
- robustez operacional, analisando o risco de falsos positivos e falsos negativos em situações reais de operação.

4.1 Desempenho Global do Sistema

A Tabela 6 apresenta uma visão consolidada dos três classificadores especializados. Cada modelo foi treinado de forma independente, respeitando as características estatísticas da sua classe e utilizando um conjunto comum de 30 atributos derivados dos perfis temporais de potência.

Tabela 6 – Resumo consolidado das métricas de desempenho dos modelos.

Modelo	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Detector de Zerada	100.00%	100.00%	100.00%	1.0000
Detector de Subperformance	95.22%	80.00%	63.16%	0.7059
Detector de Sombreamento*	95.36%	65.64%	62.44%	0.6368

Fonte: Elaboração do autor. *Média de 5 dobras (folds).

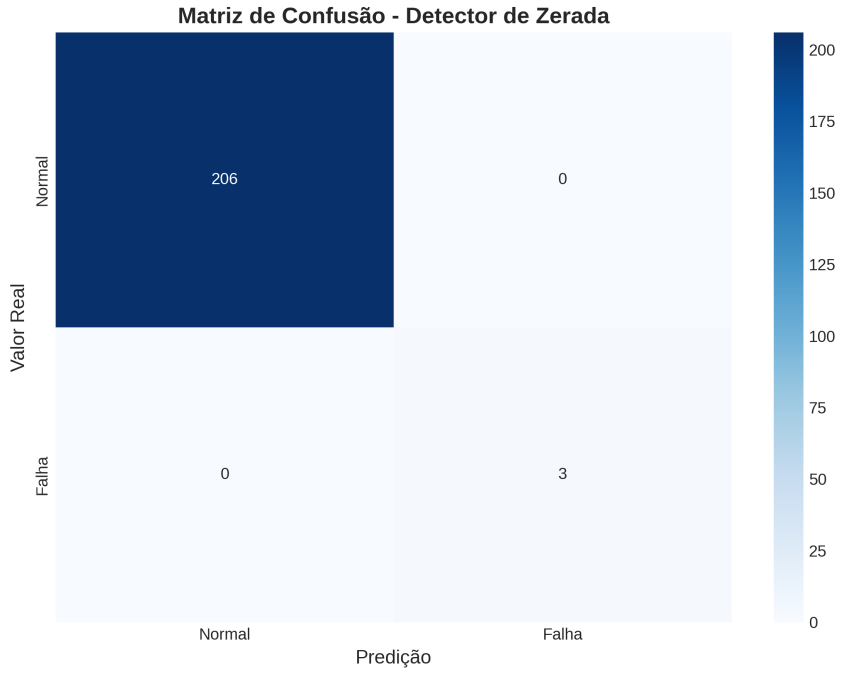
O desempenho global mostra que o sistema é capaz de detectar três tipos de falhas distintas, cada uma com comportamentos muito diferentes entre si. A classe Zerada, por exemplo, é praticamente determinística, enquanto Subperformance e Sombreamento apresentam fronteiras de decisão mais difusas, aumentando a complexidade do problema de classificação.

Além disso, observa-se que os valores de F1-Score são coerentes com a expectativa teórica: quanto mais sutil é a assinatura física da falha, mais desafiadora se torna a tarefa para o modelo.

4.2 Resultados: Detector de Stringbox Zerada

O classificador de Zerada obteve acurácia perfeita no conjunto de validação, como mostra a matriz de confusão da Figura 25. Esse resultado não é surpreendente, pois uma stringbox zerada apresenta uma assinatura operacional extremamente distinta — a ausência total de geração ao longo do dia.

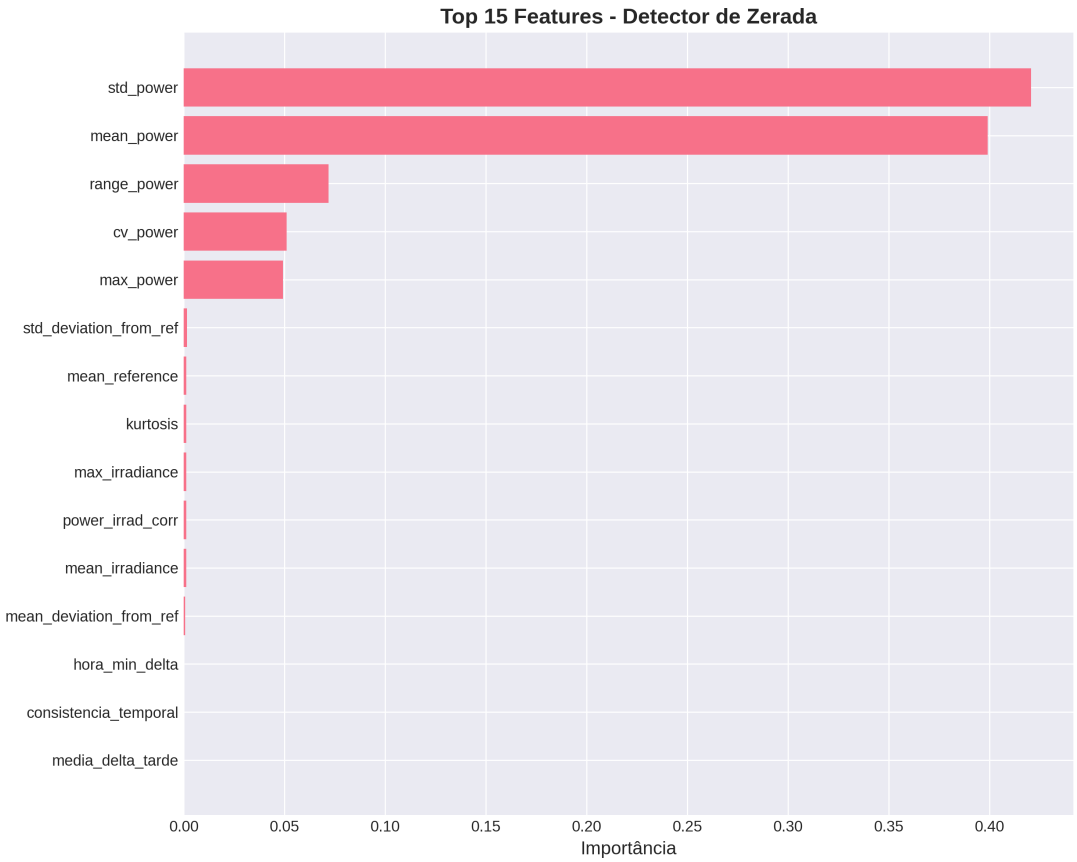
Figura 25 – Matriz de Confusão do Detector de Zerada.



Fonte: Elaboração do autor.

A análise da importância das variáveis reforça essa percepção. A Figura 26 evidencia que as variáveis associadas à estatística básica da potência dominam as decisões do modelo, com destaque para o desvio padrão da potência (*std_power*) e a potência média (*mean_power*). Esse resultado indica que o modelo identifica a condição de zerada principalmente a partir da redução do nível médio de geração e da diminuição da variabilidade da potência ao longo do dia, características típicas de falhas que resultam em produção nula ou quase nula.

Figura 26 – Importância das variáveis para o modelo de Zerada. Observa-se a predominância das variáveis associadas à estatística da potência, com destaque para o desvio padrão (*std_power*) e a potência média (*mean_power*).



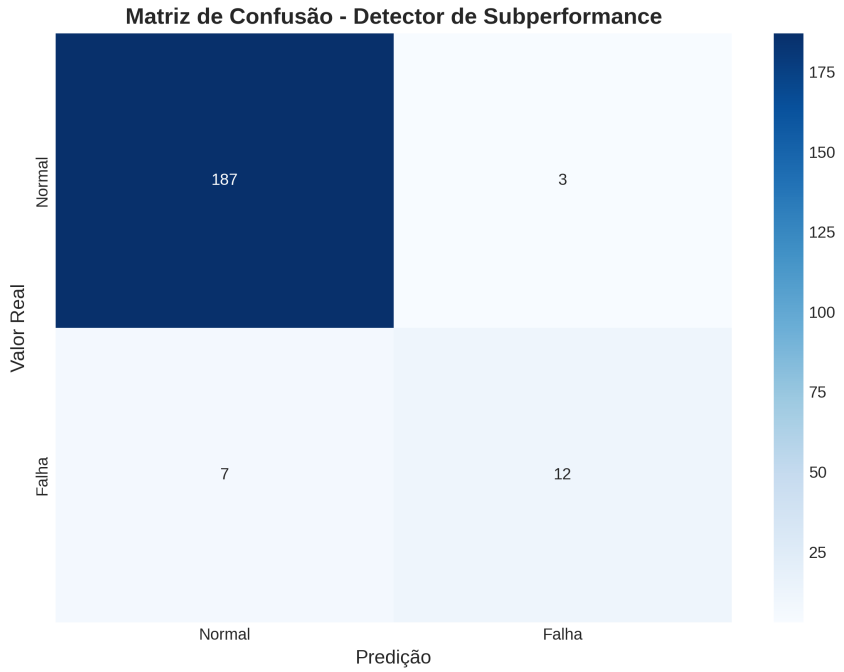
Fonte: Elaboração do autor.

A forte dependência dessa variável também demonstra que o modelo aprendeu exatamente o comportamento físico da falha. Na prática operacional, *stringboxes* zeradas representam uma condição severa que exige intervenção imediata, o que reforça a importância de se ter um classificador extremamente preciso nessa classe. A ausência de falsos negativos é particularmente relevante: perder a detecção dessa falha pode resultar em perdas energéticas significativas.

4.3 Resultados: Detector de Subperformance

A detecção de subperformance apresentou um F1-Score de 0.7059, com recall moderado. A matriz de confusão da Figura 27 mostra que o modelo adota um comportamento mais conservador, priorizando precisão em detrimento do recall.

Figura 27 – Matriz de Confusão do Detector de Subperformance.

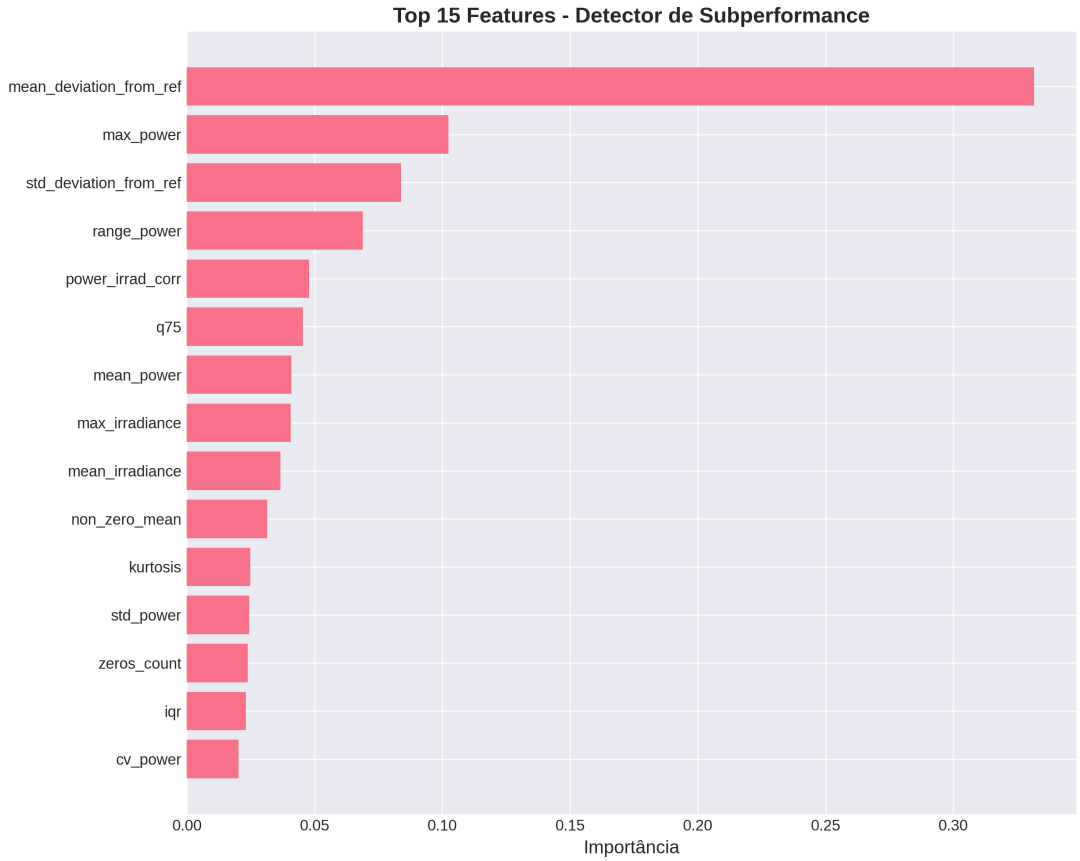


Fonte: Elaboração do autor.

Esse padrão de comportamento é adequado para ambientes operacionais onde falsos positivos podem gerar diagnósticos incorretos ou mobilização desnecessária de equipes de manutenção.

A Figura 28 mostra que a variável mais influente foi *mean_deviation_from_ref*. Essa observação é coerente: a subperformance é caracterizada por uma redução mais ou menos uniforme da curva de potência ao longo do dia, sem deformações significativas no seu formato.

Figura 28 – Importância das variáveis para o modelo de Subperformance.



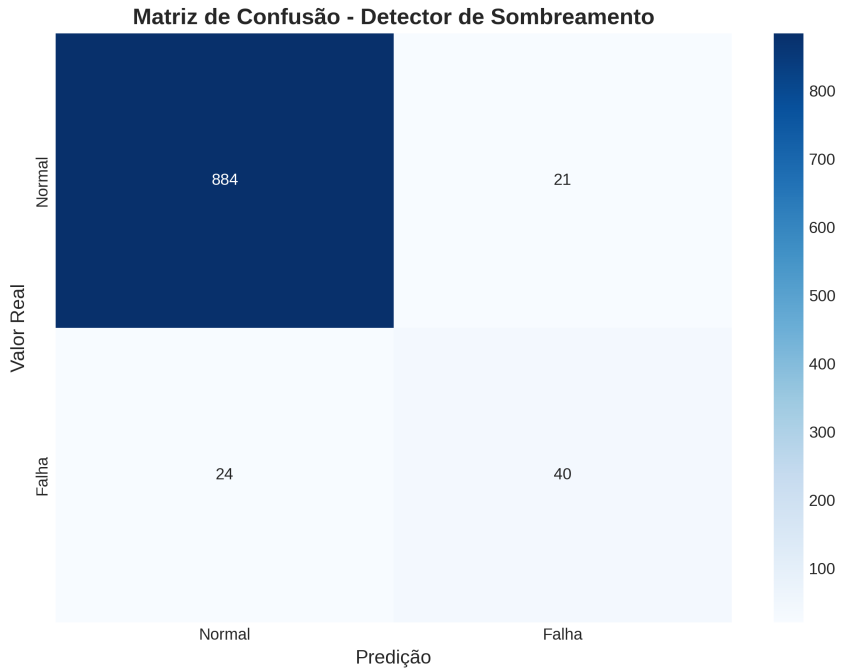
Fonte: Elaboração do autor.

Do ponto de vista físico, isso também confirma que o modelo está aprendendo comportamentos coerentes: ao contrário do sombreamento, que afeta predominantemente partes específicas do dia (manhã ou tarde), a subperformance tende a impactar toda a curva de forma mais uniforme.

4.4 Resultados: Detector de Sombreamento

O detector de Sombreamento foi avaliado via validação cruzada estratificada, devido ao baixo número de exemplos positivos. A matriz de confusão acumulada da Figura 29 mostra que o modelo possui capacidade de identificar padrões característicos da falha, embora com maior variabilidade entre os *folds*.

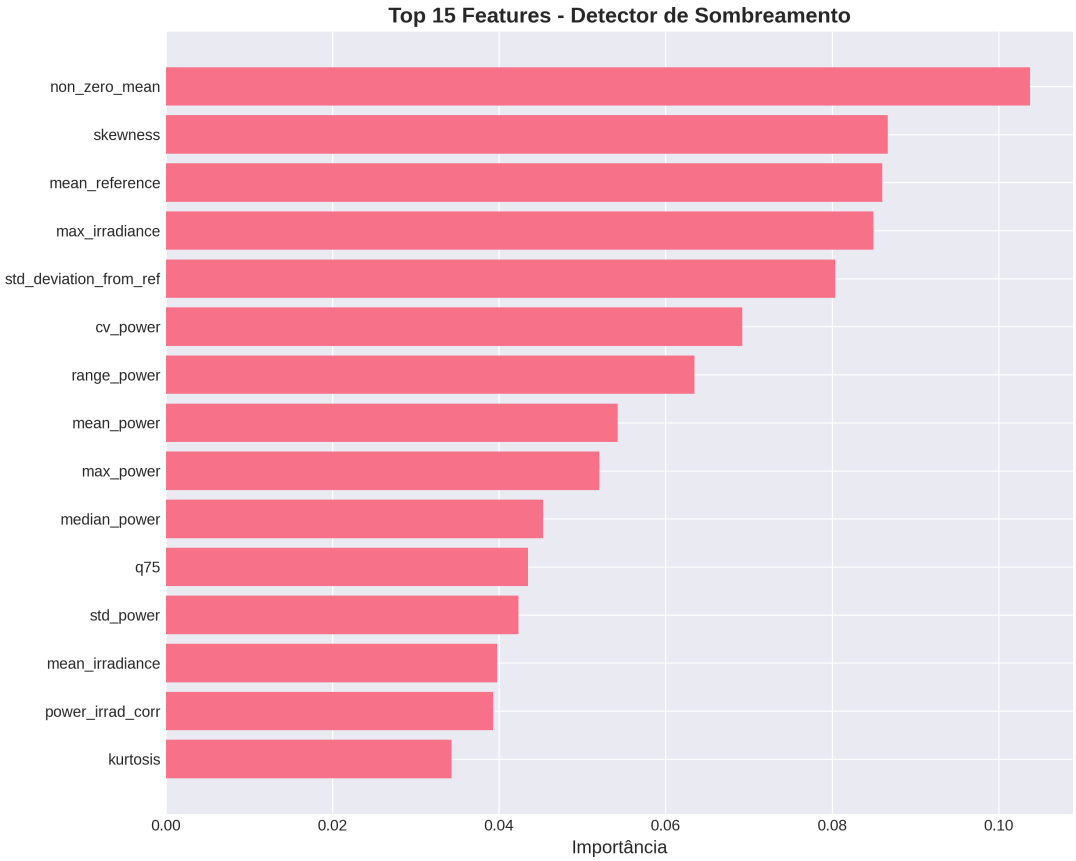
Figura 29 – Matriz de Confusão acumulada do Detector de Sombreamento.



Fonte: Elaboração do autor.

A análise das importâncias de atributos (Figura 30) mostra que, ao contrário do caso de Subperformance, o modelo depende fortemente de variáveis relacionadas à forma da curva, como assimetria (*skewness*) e razão manhã/tarde. Isso reforça a ideia de que o sombreamento não se manifesta apenas como uma queda de potência, mas sim como uma distorção localizada na curva.

Figura 30 – Importância das variáveis para o modelo de Sombreamento. Note a relevância de atributos de forma (*skewness*) e temporais (*ratio*).



Fonte: Elaboração do autor.

Em termos operacionais, tal comportamento é extremamente desejável: a identificação de sombreamento exige compreender não apenas o quanto a potência foi reduzida, mas *quando* essa redução ocorreu ao longo do dia. O modelo demonstrou essa capacidade.

4.5 Análise Qualitativa de Casos Reais

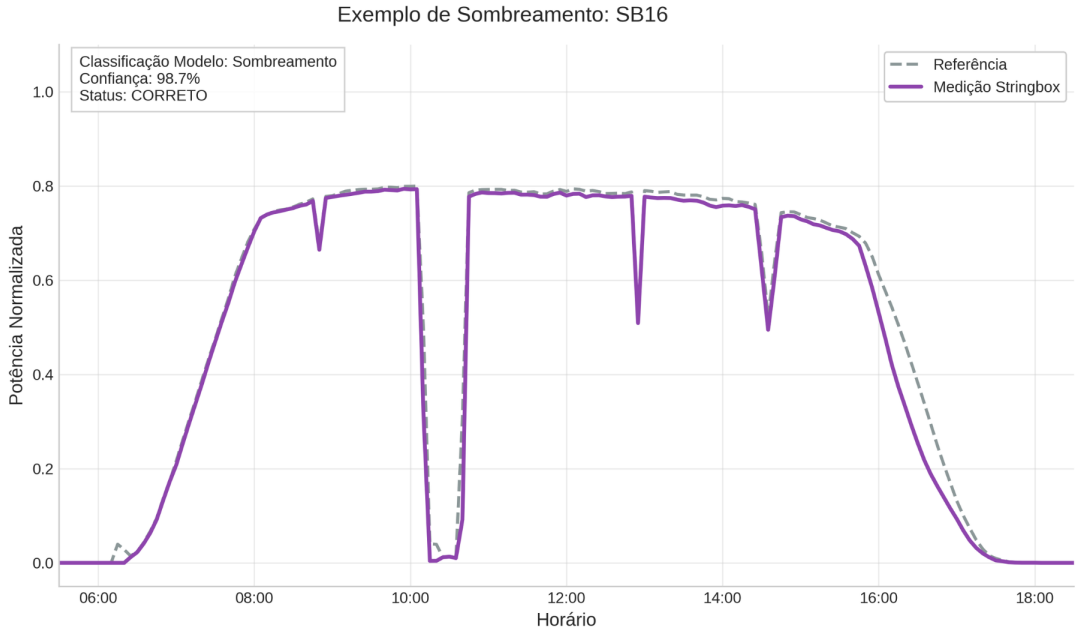
Com o objetivo de complementar a avaliação quantitativa apresentada nas seções anteriores, esta seção apresenta exemplos reais de curvas de potência analisadas pelo sistema proposto. A inclusão dessas análises permite verificar se as decisões tomadas pelos modelos são coerentes do ponto de vista físico e operacional, além de tornar o processo de classificação mais interpretável.

Para cada tipo de falha — Zerada, Subperformance e Sombreamento — é apresentado um caso representativo contendo a curva de potência normalizada da *stringbox* analisada, a curva de referência estimada e o veredito final emitido pelo modelo.

4.5.1 Análise de Caso: Sombreamento Parcial

A Figura 31 apresenta um exemplo real de *stringbox* classificada como sombreamento pelo sistema, referente à *stringbox* SB16. Observa-se que a curva de potência medida apresenta reduções abruptas e localizadas ao longo do dia, enquanto a curva de referência mantém o comportamento esperado para condições normais de operação.

Figura 31 – Exemplo de sombreamento detectado pelo sistema: SB16.



Fonte: Elaboração do autor.

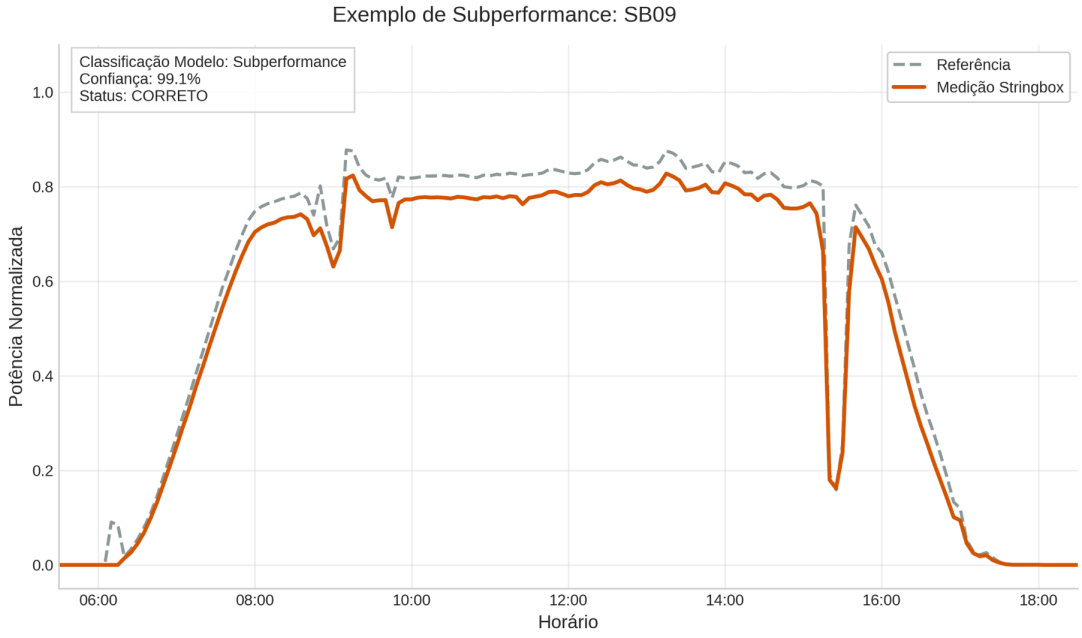
Nota-se que as quedas de potência não ocorrem de forma uniforme, mas concentram-se em intervalos específicos do dia, caracterizando uma deformação assimétrica da curva. Esse comportamento é típico de sombreamentos causados por obstáculos fixos ou móveis, como estruturas, vegetação ou sombras projetadas por elementos do próprio arranjo fotovoltaico.

O modelo classificou corretamente o evento como sombreamento, atribuindo uma confiança de 98.7%, o que indica elevada segurança na decisão. Esse resultado evidencia que o classificador não se baseia apenas em métricas globais de potência média, mas sim em atributos capazes de capturar a forma temporal da curva, como assimetria (*skewness*) e razões entre períodos do dia.

4.5.2 *Análise de Caso: Subperformance*

A Figura 32 ilustra um exemplo de subperformance identificado na *stringbox* SB09. Diferentemente do sombreamento, observa-se que a curva de potência medida mantém formato semelhante ao da referência, porém deslocada para níveis inferiores ao longo de praticamente todo o período de geração.

Figura 32 – Exemplo de subperformance detectada pelo sistema: SB09.



Fonte: Elaboração do autor.

Esse padrão é característico de falhas que reduzem a eficiência global da *stringbox*, como degradação de módulos, perdas resistivas elevadas ou falhas parciais de componentes. A ausência de deformações acentuadas na curva reforça a distinção entre subperformance e falhas de natureza temporal localizada.

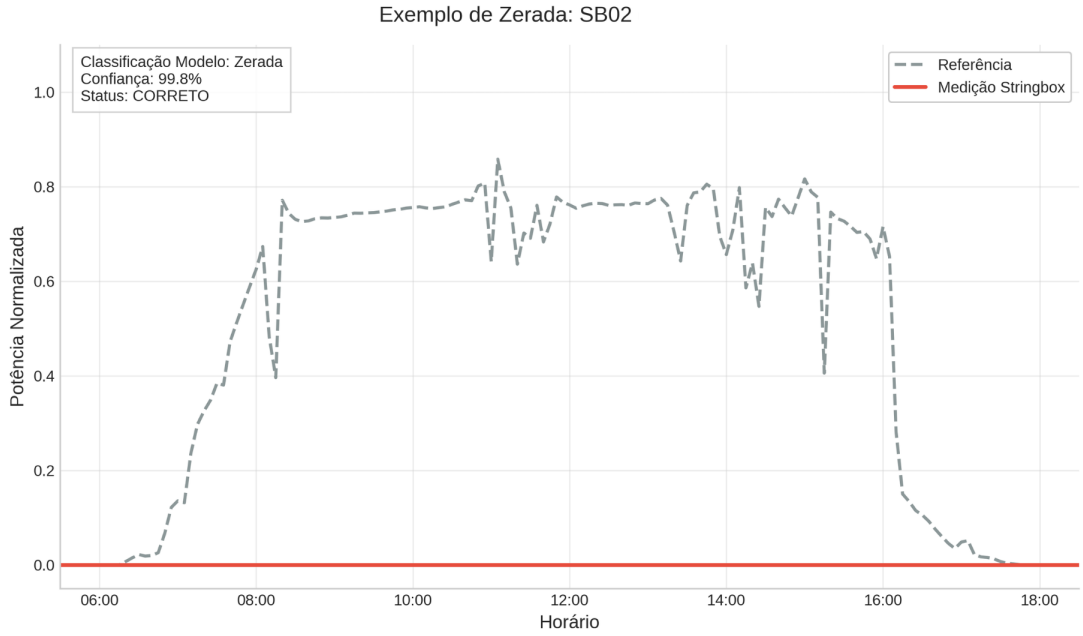
O classificador atribuiu a esta amostra uma confiança de 99.1%, classificando-a corretamente como subperformance. A decisão está alinhada com a dominância da variável `mean_deviation_from_ref`, que mede o desvio médio em relação à curva de referência e representa um indicador físico direto de perda uniforme de desempenho.

4.5.3 *Análise de Caso: Stringbox Zerada*

A Figura 33 apresenta um exemplo extremo de falha, correspondente a uma *stringbox* zerada (SB02). Observa-se que a potência medida permanece nula durante todo o período de

geração, enquanto a curva de referência indica que haveria produção esperada naquele dia.

Figura 33 – Exemplo de *stringbox* zerada detectada pelo sistema: SB02.



Fonte: Elaboração do autor.

Esse tipo de falha possui assinatura determinística, sendo facilmente identificável por atributos simples, como a proporção de medições iguais a zero. O modelo classificou corretamente o evento com confiança de 99.8%, confirmando a robustez do classificador para falhas críticas.

Do ponto de vista operacional, a correta identificação desse tipo de falha é essencial, uma vez que *stringboxes* zeradas representam perdas energéticas severas e demandam intervenção imediata.

4.6 Discussão Geral

Globalmente, os resultados validam a eficácia da arquitetura *one-vs-all* na segregação de distintas classes de falhas. Embora compartilhem a mesma base de atributos, os três modelos convergiram para regiões de decisão distintas, alinhando-se à natureza física específica de cada anomalia.

O classificador de Zerada atingiu desempenho ideal, enquanto o de Subperformance apresentou um equilíbrio consistente entre sensibilidade e precisão. Já o modelo de Sombreamento, apesar do baixo volume de exemplos positivos, foi capaz de capturar padrões temporais característicos.

Não obstante, ressaltam-se certas limitações intrínsecas ao estudo:

(i) o desbalanceamento entre classes, que impôs desafios específicos ao modelo de Sombreamento;

(ii) a janela temporal restrita de dados rotulados, o que limita a capacidade de generalização para casos atípicos;

(iii) a necessidade de expandir a engenharia de atributos, incorporando parâmetros físicos adicionais — como inclinação estimada e impacto angular — para refinar o desempenho futuro.

Ainda assim, os resultados obtidos mostram-se coerentes com a literatura e corroboram a viabilidade técnica de modelos baseados em *gradient boosting* para o diagnóstico automatizado de falhas em usinas fotovoltaicas.

5 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho atingiu seu objetivo geral ao desenvolver e validar um sistema automatizado para detecção de anomalias em *stringboxes* de usinas fotovoltaicas *utility-scale*, utilizando o algoritmo XGBoost. A abordagem baseada em aprendizado supervisionado, aliada a uma estratégia de engenharia de atributos focada na morfologia das curvas de potência, demonstrou ser uma alternativa viável e eficiente frente às limitações dos sistemas SCADA tradicionais baseados em limiares fixos.

Os resultados obtidos confirmam a hipótese de que diferentes tipos de falhas possuem assinaturas estatísticas e temporais distintas, que podem ser aprendidas por modelos de *machine learning* mesmo sem o uso de sensores de irradiação no plano do arranjo para cada *stringbox*. A estratégia de modelagem *One-vs-All*, treinando classificadores binários independentes para cada tipo de anomalia, provou-se eficaz ao permitir o diagnóstico de falhas simultâneas e facilitar a manutenção evolutiva do sistema.

Especificamente, o detector de **Stringbox Zerada** apresentou desempenho ideal, alcançando 100% de acurácia, precisão e *recall*. Este resultado valida a capacidade do modelo em identificar paradas críticas de geração, garantindo que perdas severas sejam notificadas imediatamente à equipe de operação e manutenção (O&M).

O detector de **Subperformance** demonstrou robustez com uma acurácia de 95,22% e um *F1-Score* de 0,7059. A análise de importância das variáveis revelou que o desvio médio em relação à referência (*mean_deviation_from_ref*) foi o atributo determinante, confirmando que este tipo de falha se manifesta predominantemente como uma perda de eficiência constante ao longo do dia, distinta de variações transientes.

Em relação ao **Sombreamento**, o sistema foi capaz de capturar a natureza temporal da falha, utilizando atributos de forma como a assimetria (*skewness*) e a razão de desempenho entre períodos (manhã/tarde). Embora tenha apresentado métricas ligeiramente inferiores devido ao desbalanceamento de classes (*F1-Score* de 0,6368), o modelo conseguiu distinguir com sucesso as distorções localizadas na curva de potência, validando a engenharia de atributos proposta.

5.1 Trabalhos Futuros

Considerando as delimitações deste estudo e as oportunidades identificadas durante o desenvolvimento, sugerem-se as seguintes linhas de pesquisa para a continuidade do trabalho:

- **Expansão da Base de Dados e Rotulagem:** A principal limitação encontrada foi a escassez de exemplos rotulados de sombreamento. Recomenda-se a ampliação do *dataset* histórico e o uso de técnicas de *Data Augmentation* ou aprendizagem semissupervisionada para melhorar a generalização do modelo nesta classe.
- **Integração com Dados de *Trackers*:** A incorporação de variáveis de estado dos rastreadores solares (ângulo de inclinação real, alarmes de motor) poderia refinar a detecção de anomalias, permitindo distinguir entre sombreamento por obstáculo fixo e sombreamento mútuo causado por falha no *backtracking*.
- **Implementação em *Edge Computing*:** Investigar a viabilidade de embarcar os modelos treinados diretamente nos concentradores de dados ou nas próprias *stringboxes* inteligentes, reduzindo a latência de detecção e a dependência de conectividade contínua com o servidor central.
- **Análise Econômica:** Desenvolver um módulo complementar que quantifique a perda financeira estimada para cada falha detectada, auxiliando a priorização das ordens de serviço de manutenção com base no retorno sobre o investimento (ROI).

REFERÊNCIAS

ABSOLAR. **Dados consolidados de geração fotovoltaica no Brasil**. 2025. https://www.absolar.org.br. Acesso em: 18 fev. 2025.

AGENCY, I. E. **Snapshot of Global PV Markets 2020**. 2020. IEA PVPS Report. https://www.iea.org.

ANEEL. **Resolução Normativa ANEEL nº 482/2012**. 2012. https://www.aneel.gov.br/cedoc/ren2012482.pdf. Acesso em: 18 out. 2025.

BECQUEREL, A.-E. Mémoire sur les effets électriques produits sous l'influence des rayons solaires. **Comptes Rendus de l'Académie des Sciences**, v. 10, p. 448–450, 1839.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. A. **Classification and Regression Trees**. [*S. l.*]: CRC press, 1984.

BREUNIG, M. M.; KRIEGEL, H.-P.; NG, R. T.; SANDER, J. Lof: identifying density-based local outliers. In: **Proceedings of the 2000 ACM SIGMOD international conference on Management of data**. Dallas, TX, USA: ACM, 2000. p. 93–104.

CHAPIN, D. M.; FULLER, C. S.; PEARSON, G. L. A new silicon p-n junction photocell for converting solar radiation into electrical power. **Journal of Applied Physics**, AIP, v. 25, p. 676–677, 1954.

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.*: *s. n.*], 2016. p. 785–794.

CHEN, Z.; CHEN, Y.; WU, L.; CHENG, S.; LIN, P. A deep residual neural network for faults classification in photovoltaic arrays. **Optik**, Elsevier, v. 183, p. 295–304, 2019. Abordagem baseada em Deep Learning e Curvas I-V.

DHIMISH, M.; HOLMES, V.; MEHRDADI, B.; DALES, M. Photovoltaic degradation rate analysis utilizing a developed novel algorithm. **Energy Conversion and Management**, Elsevier, v. 176, p. 112–121, 2018. Abordagem baseada em análise estatística e limiares.

DOMINGOS, P. A few useful things to know about machine learning. **Communications of the ACM**, ACM New York, NY, USA, v. 55, n. 10, p. 78–87, 2012.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural networks**, Elsevier, v. 2, n. 5, p. 359–366, 1989.

IEC. **IEC 61724-1: Photovoltaic system performance - Part 1: Monitoring**. Geneva, 2017.

IRENA. **Renewable Capacity Statistics 2024**. 2024. International Renewable Energy Agency. Disponível em: https://www.irena.org/Publications/2024/Mar/Renewable-Capacity-Statistics-2024.

JAPKOWICZ, N.; SHAH, M. **Evaluating learning algorithms: a classification perspective**. [*S. l.*]: Cambridge University Press, 2011.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: IEEE. **2008 Eighth IEEE International Conference on Data Mining**. Pisa, Italy, 2008. p. 413–422.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MELLIT, A.; KALOGIROU, S. A. Artificial intelligence techniques for photovoltaic applications: A review. **Progress in Energy and Combustion Science**, Elsevier, v. 34, n. 5, p. 574–632, 2008.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Cambridge, MA: MIT press, 2012. (Adaptive Computation and Machine Learning series).

NREL. **Photovoltaic System Cost Benchmark: Q1 2020**. 2021. National Renewable Energy Laboratory. https://www.nrel.gov.

PIERRO, M.; BUCCI, F.; FELICE, M. D.; CORNARO, C.; MOSER, D.; PEROTTO, M. Data-driven up-scaling of pv power generation from a reference pv plant to a regional capacity. **Solar Energy**, Elsevier, v. 189, p. 315–326, 2019. Uso de Machine Learning para previsão em larga escala.

PINHO, J. T.; GALDINO, M. A. **Manual de Engenharia para Sistemas Fotovoltaicos**. Edição revisada e atualizada. Rio de Janeiro: CEPEL / CRESESB, 2014. Disponível em: http://www.cresesb.cepel.br.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, n. 6, p. 306–408, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of research and development**, IBM, v. 3, n. 3, p. 210–229, 1959.

VILLALVA, M. G.; GAZOLI, J. R. **Energia Solar Fotovoltaica: Conceitos e Aplicações**. 2. ed. São Paulo: Érica, 2015. ISBN 978-8536504162.

APÊNDICE A – EXEMPLO DO ARQUIVO DE DADOS JSON USADO PARA O TREINAMENTO DOS MODELOS

O Código 1 apresenta um trecho ilustrativo do arquivo JSON utilizado no treinamento dos modelos propostos, com omissão de registros intermediários para fins de visualização.

Código-fonte 1 – Trecho ilustrativo do arquivo JSON de entrada utilizado no treinamento dos modelos.

```
{
  "irradiance_col": "irradiance",
  "string-box": [
    {
      "sample_time": "2025-05-21T00:00:00",
      "string_box_id": 5201,
      "string_box": "Plant G - Inverter A - SB01",
      "inverter_id": 52,
      "normalized_power_dc": null,
      "reference_sb_power": null,
      "irradiance": null
    },
    {
      "sample_time": "2025-05-21T00:05:00",
      "string_box_id": 5201,
      "string_box": "Plant G - Inverter A - SB01",
      "inverter_id": 52,
      "normalized_power_dc": null,
      "reference_sb_power": null,
      "irradiance": null
    },

    // ... registros intermediarios omitidos

    {
      "sample_time": "2025-05-21T07:30:00",
      "string_box_id": 5201,
      "string_box": "Plant G - Inverter A - SB01",
      "inverter_id": 52,
      "normalized_power_dc": 0.5134224036163019,
      "reference_sb_power": 0.5243069284936343,
      "irradiance": 273.06872507731117
    },

    // ... registros intermediarios omitidos

    {
```

```
      "sample_time": "2025-05-21T22:25:00",
      "string_box_id": 5201,
      "string_box": "Plant G - Inverter A - SB01",
      "inverter_id": 52,
      "normalized_power_dc": null,
      "reference_sb_power": null,
      "irradiance": null
    },
    {
      "sample_time": "2025-05-21T22:30:00",
      "string_box_id": 5201,
      "string_box": "Plant G - Inverter A - SB01",
      "inverter_id": 52,
      "normalized_power_dc": null,
      "reference_sb_power": null,
      "irradiance": null
    }
  ]
}
```