INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÂO EM ENGENHARIA MECATRÔNICA

LUCAS MOINO ARMADA

Detecção de Anomalias em Instalação Fotovoltaica Utilizando Aprendizado de Máquina.

FLORIANÓPOLIS, 2024.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL MECÂNICA
CURSO DE GRADUAÇÂO EM ENGENHARIA MECATRÔNICA

LUCAS MOINO ARMADA

Detecção de Anomalias em Instalação Fotovoltaica Utilizando Aprendizado de Máquina.

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Engenheiro em Mecatrônica.

Orientador:
Prof. Delcino Picinin Junior, Doutor

FLORIANÓPOLIS, 2024.

Ficha de identificação da obra elaborada pelo autor.

Armada, Lucas
Detecção de Anomalias em Instalação Fotovoltaica Utilizando **Aprendizado de Máquina / Lucas Armada;** orientação **de Delcino Junior. – Florianópolis, SC,** 2024.
64 p.

Trabalho de Conclusão de Curso (TCC) – Instituto Federal **de Santa Catarina, Câmpus Florianópolis.** Bacharelado **em Engenharia Mecatrônica.** Departamento **Acadêmico de Metal** Mecânica.
Inclui Referências.

1. Machine learning. 2. Aprendizado não supervisionado. **3. Análise de dados. 4.** Detecção de anomalias. **I. Junior, Delcino. II. Instituto Federal de Santa** Catarina. **III. Detecção de Anomalias em** Instalação Fotovoltaica **Utilizando Aprendizado de** Máquina.

DETECÇÃO DE ANOMALIAS EM INSTALAÇÃO FOTOVOLTAICA UTILIZANDO APRENDIZADO DE MÁQUINA

LUCAS MOINO ARMADA

Este trabalho foi julgado adequado para obtenção do título de Engenheiro em Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 27 de setembro, 2024.

Banca Examinadora:

Documento assinado digitalmente
gov.br
DELCINO PICININ JUNIOR
Data: 25/09/2024 13:12:10-0300
Verifique em https://validar.iti.gov.br

Delcino Picinin Junior, Dr.

Documento assinado digitalmente
gov.br
MAURICIO EDGAR STIVANELLO
Data: 24/09/2024 17:34:40-0300
Verifique em https://validar.iti.gov.br

Mauricio Edgar Stivanello, Dr.

IFSC

Documento assinado digitalmente
gov.br
BRUNO ALBERTO PERUCHI
Data: 25/09/2024 09:57:54-0300
Verifique em https://validar.iti.gov.br

Bruno Alberto Peruchi, Ms.

IFSC

RESUMO

A demanda por energias renováveis vem crescendo a cada ano no Brasil e no mundo. Nesse sentido, dentre as alternativas para fazer frente aos combustíveis fósseis a energia fotovoltaica tem se mostrado uma das mais atraentes, principalmente no Brasil tendo em vista a alta incidência solar em nosso território. Nas matrizes fotovoltaicas, os indícios ou sinais de possíveis falhas podem não ser identificadas a tempo de impactarem todo o sistema ou de prejudicarem sua eficácia. A solução encontrada para auxiliar na detecção destes indícios de possíveis falhas e problemas parte da abordagem da análise de conjuntos de dados de medições de um sistema fotovoltaico com e sem falhas injetadas manualmente, partindo do método de aprendizado de máquinas não supervisionado, detectar anomalias que antecedem falhas comuns utilizando dados alimentados à um modelo e sem necessidade de conhecimento contextual. Este trabalho propõe agregar conhecimentos de engenharia mecatrônica com a área abrangente de energia fotovoltaica para a concepção de um sistema que, em poucos segundos, consiga detectar anomalias e auxiliar profissionais em possíveis diagnósticos de falhas, entender e identificar quedas de desempenho, alertar sobre falhas em componentes que levariam à impactos no sistema, realizar manutenção preditiva, entre outros. Durante este trabalho é demonstrado o processo de análise dos dados e classificação das anomalias, tratamento e filtragem dos sinais até o desenvolvimento da detecção de anomalias. Os resultados encontrados foram de taxas de precisão de até 100% com um modelo simples nas detecções de anomalias de densidade global em algumas falhas; porém mesmo com inserção de uma extensão do modelo utilizado, não houve um grande salto nos resultados para anomalias de densidade local. No final, conclui-se que a versão padrão de *Isolation Forest* é mais robusta que a sua versão extendida, lidando melhor com desafios como alta dimensionalidade e quantidade dos dados, porém mantendo uma taxa de detecção mínima de anomalias de densidade local, em torno dos 50%.

Palavras-chave: Machine Learning, Análise de Dados, Energia Fotovoltaica, Aprendizado não Supervisionado, Deteção de Anomalias.

ABSTRACT

The demand for renewable energy has been growing every year in Brazil and around the world. Among the alternatives to fossil fuels, photovoltaic energy has proven to be one of the most attractive options, particularly in Brazil due to the high solar incidence across the country. In photovoltaic systems, early signs or indicators of potential failures may not be identified in time, which can impact the entire system or hinder its effectiveness. The solution proposed to assist in the detection of these potential failure signs involves the analysis of measurement data from a photovoltaic system, both with and without manually injected faults. By utilizing an unsupervised machine learning approach, this method aims to detect anomalies that precede common failures without requiring contextual knowledge. This work aims to combine mechatronics engineering knowledge with the broad field of photovoltaic energy to design a system capable of detecting anomalies within seconds. This system will aid professionals in diagnosing potential failures, understanding and identifying performance drops, alerting to component failures that could impact the system, enabling predictive maintenance, and more. Throughout this work, the process of data analysis and anomaly classification, as well as the treatment and filtering of signals up to the development of anomaly detection, is demonstrated. The results found were precision rates of up to 100% with a simple model in global density anomaly detection for some faults; however, even with the addition of an extension to the model used, there was no significant improvement in the results for local density anomalies. In the end, it was concluded that the standard version of *Isolation Forest* is more robust than its extended version, handling challenges such as high dimensionality and data volume better, while maintaining a minimum detection rate for local density anomalies, around 50%.

Keywords: Machine Learning, Data Analysis, Photovoltaic Energy, Unsupervised Learning, Anomaly Detection.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Aprendizagem de máquina (*machine learning*) emerge como uma vertente revolucionária na esfera da inteligência artificial, desempenhando um papel crucial na forma como os dados são analisados e utilizados no século XXI. Este campo interdisciplinar combina métodos da estatística, ciência da computação e engenharia de dados para capacitar computadores a aprenderem e tomarem decisões baseadas em padrões e inferências derivados de grandes conjuntos de dados.

A evolução do *machine learning* está intrinsecamente ligada ao aumento exponencial da capacidade computacional e à disponibilidade de grandes volumes de dados. Algoritmos de aprendizagem de máquina são treinados usando conjuntos de dados para reconhecer padrões e características, os quais são posteriormente aplicados para fazer previsões ou tomar decisões em novos dados. Este processo permite uma variedade de aplicações, desde a filtragem de e-mails indesejados até o desenvolvimento de sistemas autônomos de veículos.

Na área de energia, a inteligência artificial vem sendo cada vez mas utilizada, conforme relato da Agência Internacional da Energia em seu artigo *Why AI and energy are the new power couple*.

> Os sistemas de energia estão se tornando muito mais complexos à medida que a demanda por eletricidade cresce e os esforços de descarbonização aumentam. No passado, as redes direcionavam energia de usinas de energia centralizadas. Agora, os sistemas de energia precisam cada vez mais apoiar fluxos multidirecionais de eletricidade entre geradores distribuídos, a rede e os usuários. O crescente número de dispositivos conectados à rede, desde estações de carregamento de veículos elétricos (VE) até instalações solares residenciais, torna os fluxos menos previsíveis (IEA, 2023).

As vantagens e as utilizações são muitas, a inteligência artifical pode gerenciar redes inteiras de usinas, deixá-las mais flexíveis prevendo demanda e suprimento, impedir falhas na rede com as suas aplicações em manutenção preditiva, ou até mesmo encontrar falhas antes de causarem algum dano real à rede, onde se concentra este estudo, através de detecção de anomalias.

Dentro deste contexto, monitorar medições de uma instalação fotovoltaica não é uma tarefa simples, principalmente quando existe a necessidade de identificar anomalias, que muitas vezes são mascaradas pela grande quantidade de variáveis e ruídos. Este estudo se concentra especificamente em entender os principais desafios desta área e em estudar alternativas para detectar essas anomalias, utilizando sistemas simples e eficientes que poderiam ser treinados rapidamente em lotes ou até de forma incremental.

2 OBJETIVO

Para orientar o projeto, estabeleceu-se um objetivo principal acompanhado de objetivos específicos, também conhecidos como secundários. Esses objetivos não apenas ajudaram a compreender os fundamentos abordados de forma ampla, mas também foram essenciais para concluir o trabalho.

2.1 Objetivo Geral

Realizar a detecção de anomalias em um conjunto de dados de medições de uma usina fotovoltaica. Esse objetivo abrange a seleção de um ou mais modelos de *machine learning* para a identificação de *outliers* nas medições e também a utilização de uma ou mais métricas de avaliação para selecionar o método mais eficiente na resolução do problema proposto.

2.2 Objetivos Específicos

a) Seleção de um banco de dados com medições de uma instalação fotovoltaica, com as mesmas abrangendo desde um funcionamento normal até o funcionamento ocorrendo os defeitos mais comuns;

b) Estudo da teoria de *machine learning*, mais especificamente os modelos de detecção de *outliers* não supervisionados;

c) Estudo a seleção de bibliotecas de filtragem de ruídos nas medições;

d) Estudo do algoritmo selecionado;

e) Selecionar uma métrica de avaliação que melhor se encaixa no contexto;

f) Validar os métodos diferentes através da avaliação;

g) Selecionar o método que performa melhor em relação a métrica e ao objetivo geral.

3 FUNDAMENTAÇÃO TEÓRICA

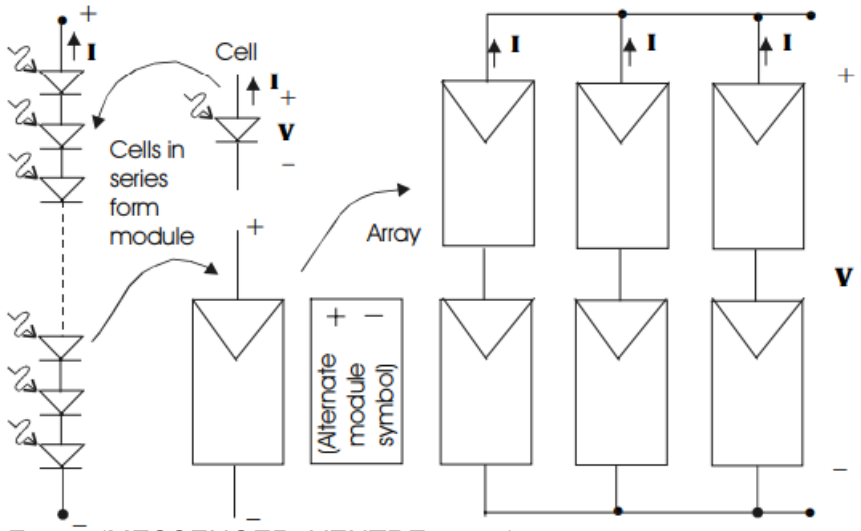
3.1 Fundamentação Teórica de um Sistema Fotovoltaico

O sistema fotovoltaico cujos os dados são utilizados neste trabalho, segundo (BAKDI *et al.*, 2021, p. 8) é um sistema fotovoltaico conectado à rede típico e com falhas reais sobre os modos *maximum power point tracker* e *intermediate power point tracker*, MPPT e IPPT respectivamente, em condições práticas.

3.1.1 Introdução

Os sistemas fotovoltaicos são projetados em torno da célula fotovoltaica. Como uma célula fotovoltaica típica produz menos de 3 watts a aproximadamente 0,5 volt dc, as células devem ser conectadas em configuração série-paralelo para produzir energia suficiente para aplicações de alta potência (MESSENGER; VENTRE, 2004, p. 47). Também de acordo com o autor, módulos fotovoltaicos podem possuir picos de potência de saída, desde 10 watts até mais de 300 watts. A figura 1 mostra como as células são configuradas em módulos e como os módulos estão conectados em um conjunto.

Figura 1 – Células Fotovoltaicas



Fonte: (MESSENGER; VENTRE, 2004).

3.1.2 Célula Fotovoltaica

De acordo com Messenger e Ventre (2004, p. 47), "a célula fotovoltaica é uma junção de pn[1] especialmente projetada ou um dispositivo de barreira Schottky".

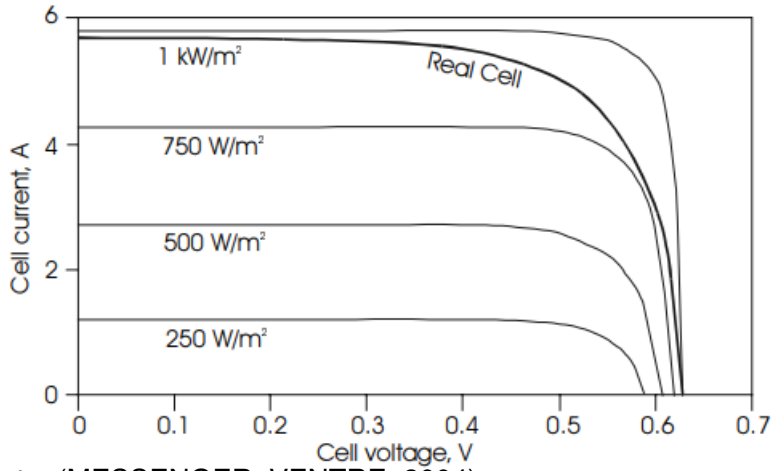
[1] A junção PN, essencial em dispositivos semicondutores, ocorre quando materiais tipo P e N entram em contato, levando à difusão de portadores de carga e à formação de uma região de depleção, onde se cria uma barreira de potencial que regula o fluxo de corrente entre as regiões, permitindo a passagem apenas sob uma tensão externa(STREETMAN, 1990, p.179-193).

Ainda segundos os mesmos autores, a equação do diodo (1) descreve a operação de uma célula fotovoltaica sombreada:

$$I = I_0(e^{\frac{qV}{kT}} - 1) \tag{1}$$

> A figura 2 descreve as características I-V de uma célula fotovoltaica. Note que as quantidades de corrente e tensão disponíveis das células dependem do nível de iluminação na mesma. A equação (1) representa a equação da característica I-V ideal, onde IL é o componente da corrente da célula devido a fótons, $q = 1,6x10 - 19coul$, $k = 1,38x10 - 23j/K$ e T é a temperatura da célula em K. Embora as características I-V das células fotovoltaicas reais diferem um pouco desta versão ideal, a equação da figura 2 fornece um meio de determinar os limites ideais de desempenho das células fotovoltaicas. (MESSENGER; VENTRE, 2004, p. 49).

Figura 2 – Características I-V Reais e Ideais da Célula Fotovoltaica Sob Diferentes Iluminações



Fonte: (MESSENGER; VENTRE, 2004).

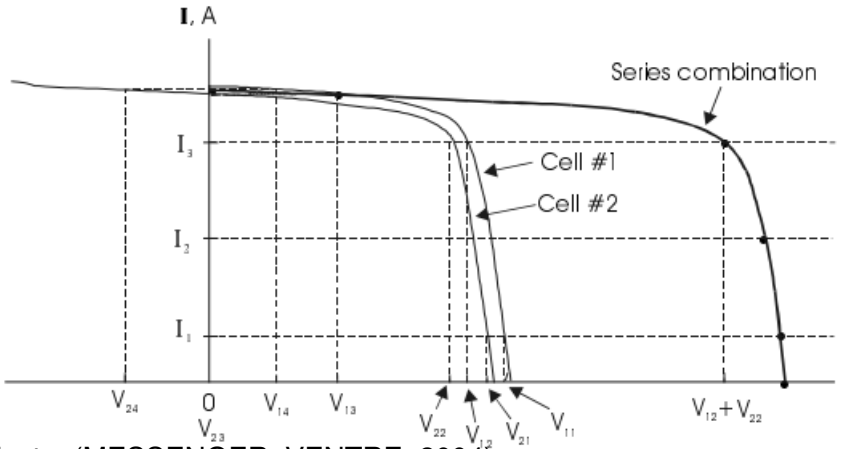
3.1.3 Módulo Fotovoltaico

Em um módulo fotovoltaico, também chamado de módulo solar, há várias células solares conectadas. A junção de vários módulos solares conectados eletricamente é chamado de painel solar, e por último, um array fotovoltaico consiste em vários painéis solares.

> Se um módulo solar é feito a partir de um conjunto de células solares, podemos conectá-las de diferentes maneiras. Primeiro, podemos conectá-las em uma conexão em série. Em uma conexão em série, as tensões se somam. Por exemplo, se a tensão de circuito aberto de uma célula for igual a 0,6 V, uma sequência de três células produzirá uma tensão de circuito aberto de 1,8 V. Para células solares conectadas em série, as correntes não se somam. Em contraste, a corrente de toda a sequência é determinada pela célula que gera a menor corrente. Portanto, a corrente total em uma sequência de células solares é igual à menor corrente gerada por uma única célula solar. (SMETS *et al.*, 2016).

Também de acordo com o autor, se as células são conectadas em paralelo, a tensão é a mesma em todas as células solares, enquanto as correntes das células solares se somam. Se conectarmos, por exemplo, células com as mesmas características I-V em paralelo, a corrente se torna três vezes maior, enquanto a tensão é a mesma que a de uma única célula. A figura 3 ilustra a curva I-V de células fotovoltaicas conectadas em séries e em paralelo.

Figura 3 – Curva I-V em Série e Paralelo



Fonte: (MESSENGER; VENTRE, 2004).

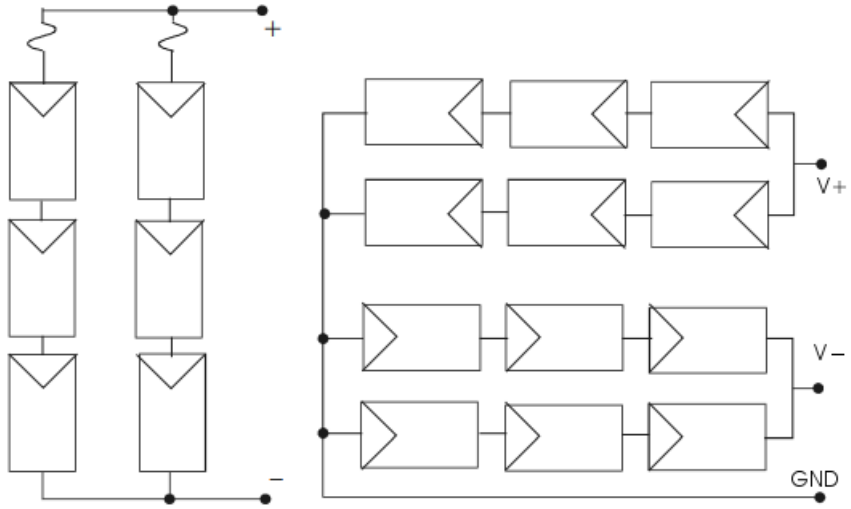
3.1.4 Matriz Fotovoltaica

Por último, o array fotovoltaico é a junção de diversos módulos solares e/ou painéis solares.

> Se forem necessárias tensões ou correntes mais altas do que as disponíveis em um único módulo, os módulos devem ser conectados em matrizes. Conexões em série resultam em tensões mais altas, enquanto conexões em paralelo resultam em correntes mais altas. Quando os módulos são conectados em série, é desejável que a produção máxima de energia de cada módulo ocorra na mesma corrente. Quando os módulos são conectados em paralelo, é desejável que a produção máxima de energia de cada módulo ocorra na mesma tensão (MESSENGER; VENTRE, 2004, p. 56).

Para igualar a corrente ou tensão em cada módulo do conjunto, é usado diodos de derivação ou um sistema de rastreamento do ponto de potência máxima (MPPT). A figura 4 abaixo ilustra matrizes em série e em paralelo.

Figura 4 – Matrizes Fotovoltaicas em Série e Paralelo



Fonte: (MESSENGER; VENTRE, 2004).

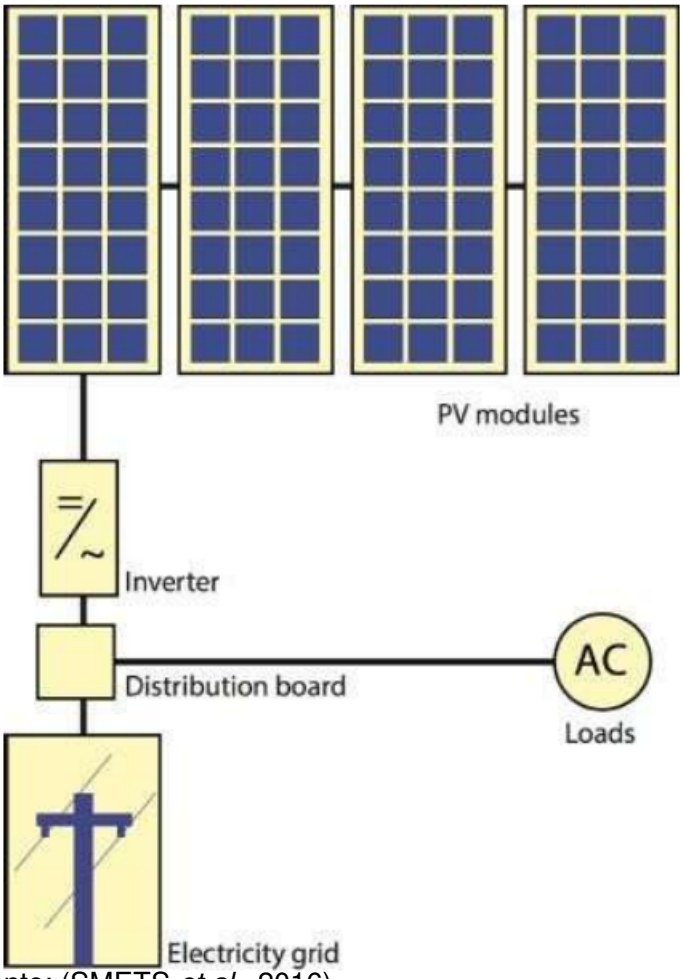
3.1.5 Sistemas Fotovoltaicos Conectados à Rede

De acordo com (SMETS *et al.*, 2016), os sistemas fotovoltaicos podem ser desde pequenos e bem simples, compostos apenas de um módulo e *load*, até grandes usinas elétricas com picos de MegaWatts. Dependendo da configuração do sistema, é possível distinguir 3 sistemas fotovoltaicos principais: stand-alone, grid-connected e híbrido.

> Sistemas fotovoltaicos conectados à rede se tornaram cada vez mais populares para aplicações no ambiente construído. Eles são conectados à rede por meio de inversores, que convertem a energia DC em eletricidade AC. Em sistemas pequenos, como os instalados em residências, o inversor é conectado ao quadro de distribuição, de onde a energia gerada pelos painéis solares é transferida para a rede elétrica ou para os aparelhos elétricos AC da casa. Em princípio, esses sistemas não requerem baterias, já que estão conectados à rede, que atua como um buffer para o qual um excesso de eletricidade gerada pelos painéis solares é transportado. A rede também fornece eletricidade para a casa em momentos de geração insuficiente de energia solar (SMETS *et al.*, 2016).

A figura 5 abaixo ilustra um sistema fotovoltaico conectado à rede.

Figura 5 – Sistema Fotovoltaico Conectado à Rede Elétrica



Fonte: (SMETS *et al.*, 2016).

3.1.6 Componentes de um Sistema Fotovoltaico

Muitos componentes são necessários para um sistema fotovoltaico funcional, segundo (SMETS *et al.*, 2016) esses componentes juntos são chamados de *balance of system* (BOS). Quais componentes são necessários depende do tipo de sistema, entretanto os mais importantes são:

a) Uma estrutura de montagem para fixar os módulos e direcioná-los para o sol;

b) Um armazenamento de energia, que é uma parte vital dos sistemas stand-alone. Geralmente, as baterias são usadas neste caso;

c) Conversores AC-DC, para converterem a saída do módulo que terá uma tensão variável dependendo da hora do dia e das condições climáticas, em uma tensão de saída compatível que possa ser usada como entrada para um inversor em um sistema conectado à rede;

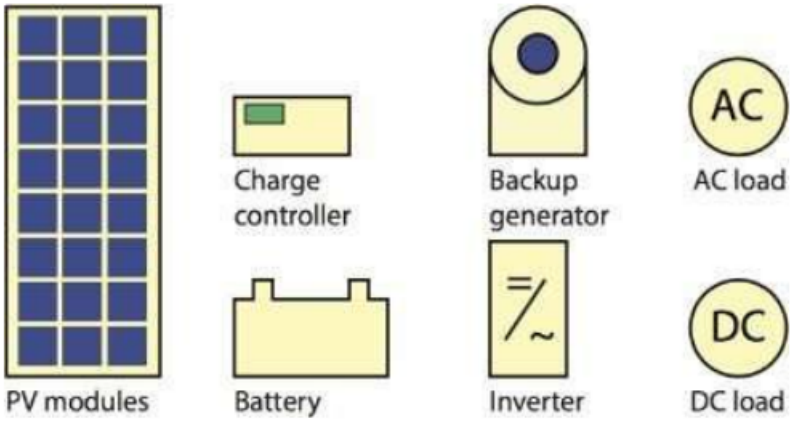
d) Inversores, que são usados em sistemas conectados à rede para conver-

ter a eletricidade DC originada pelos módulos fotovoltaicos em eletricidade AC que pode ser alimentada na rede elétrica;

e) Cabos, que são usados para conectar os diferentes componentes do sistema fotovoltaico entre si e à carga elétrica.

A figura 6 abaixo ilustra os diferentes componentes:

Figura 6 – Componentes de um Sistema Fotovoltaico



Fonte: (SMETS *et al.*, 2016).

3.1.7 MPPT e IPPT

Segundo FUCHS; MASOUM (2015) *maximum power point tracker* é o nome utilizado do método que rastreia o ponto de potência máxima (MPP) da curva característica I-V do módulo solar, ajustando a tensão de saída para garantir que ele opere sempre no MPP.

Entretanto, o MPP depende das condições ambientais. Se a irradiância ou a temperatura mudarem, as características I-V e P-V também mudarão e, portanto, a posição do MPP pode se deslocar. Portanto, as mudanças na curva I-V devem ser rastreadas continuamente para que o ponto de operação possa ser ajustado para estar no MPP após mudanças nas condições ambientais (SMETS *et al.*, 2016).

O IPPT *intermediate power point tracker* citado e utilizado pelos autores BAKDI *et al.* (2021) é equivalente ao MPPT, porém rastreando um ponto de potência intermediário no caso de potência limitada.

3.2 Fundamentos de *Machine Learning*

De acordo com GÉRON (2019, p. 4) *Machine Learning* é a ciência e a arte de programar computadores para que eles possam aprender com dados. *Machine Learning* é ótima para:

a) Problemas para os quais as soluções existentes exigem muita afinação manual ou longas listas de regras: um algoritmo de *Machine Learning* pode frequentemente simplificar o código e ter um desempenho melhor.

b) Problemas complexos para os quais não há uma boa solução usando uma abordagem tradicional: as melhores técnicas de *Machine Learning* podem encontrar uma solução.

c) Ambientes flutuantes: um sistema de *Machine Learning* pode se adaptar a novos dados.

d) Obter insights sobre problemas complexos e grandes quantidades de dados (GÉRON, 2019, p. 7).

Para explicar o que é *Machine Learning*, BURKOV (2019) diz: Vamos começar dizendo a verdade: máquinas não aprendem. O que uma "máquina de aprendizado"típica faz é encontrar uma fórmula matemática que, quando aplicada a coleção de entradas (chamadas de "dados de treinamento"), produz as saídas desejadas. Essa fórmula matemática também gera as saídas corretas para a maioria das outras entradas (diferentes dos dados de treinamento), desde que essas entradas venham da mesma ou de uma distribuição estatística semelhante àquela da qual os dados de treinamento foram retirados.

3.2.1 Introdução

Machine Learning é um subcampo da ciência da computação que se preocupa em construir algoritmos que, para serem úteis, dependem de uma coleção de exemplos de alguns fenômenos. Esses exemplos podem vir da natureza, serem criados manualmente por humanos ou gerados por outro algoritmo. *Machine Learning* também pode ser definida como o processo de resolver um problema prático por: 1) coletar um conjunto de dados e 2) construir algoritmicamente um modelo estatístico com base nesse de dados. Assume-se que esse modelo estatístico seja usado de alguma forma para resolver o problema prático (BURKOV, 2019, p. 3).

Segundo BISHOP (2006, p. 1) o desafio de procurar por padrões é fundamental e tem uma longa e vitoriosa história. O campo do reconhecimento de padrões está preocupado com a descoberta automática de regularidades em dados por meio do uso de algoritmos de computador e com o uso dessas regularidades para tomar ações, como classificar os dados em diferentes categorias.

3.2.2 Sistemas de *Machine Learning*

Existem diferentes tipos de sistemas de *Machine Learning*, tantos que GÉRON (2019) os classifica em categorias amplas, como:

a) Se são ou não treinados com supervisão humana (supervisionado, não supervisionado, semi-supervisionado e aprendizado por reforço);

b) Se podem ou não aprender incrementalmente em tempo real (aprendizado online versus em lote);

c) Se funcionam simplesmente comparando novos pontos de dados com pontos de dados conhecidos, ou se detectam padrões nos dados de treinamento e constroem um modelo preditivo, assim como os cientistas fazem (aprendizado baseado em instância versus baseado em modelo).

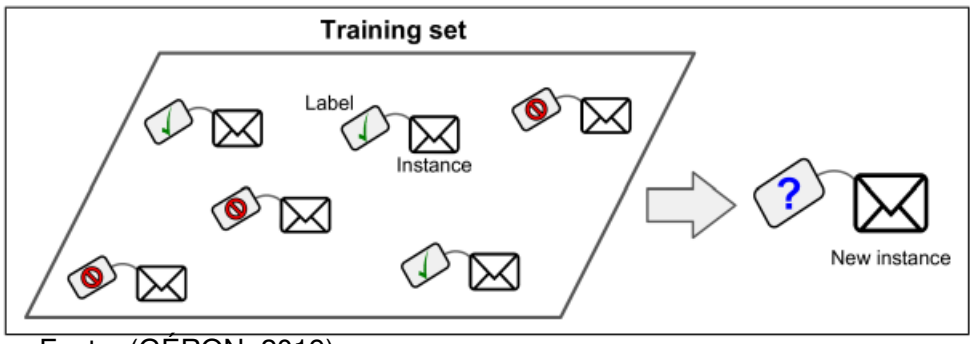
3.2.2.1 Tipos de Aprendizado

Algoritmos de *Machine Learning* são classificados pela quantidade de supervisão que recebem, neste trabalho é explicado apenas os dois tipos mais comuns, supervisionado e não supervisionado.

3.2.2.1.1 Supervisionado

Os considerados supervisionados são aqueles que recebem os dados de treinamento já com as devidas soluções, chamadas de *labels* (figura 7). Alguns dos principais tipos de algoritmos supervisionados são:

Figura 7 – Aprendizado Supervisionado



Fonte: (GÉRON, 2019).

a) Regressão Linear;

b) Regressão Logística;

c) K-Nearest neighbor;

d) Support Vector Machine;

e) Árvores de Decisão;

f) Bagging;

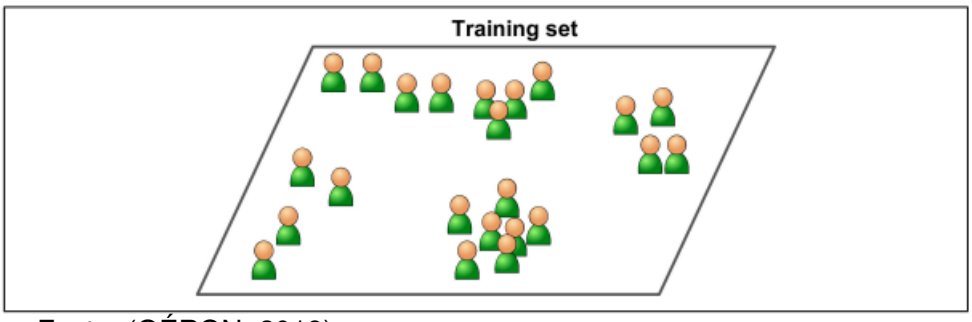
g) Boosting;

h) Redes Neurais.

3.2.2.1.2 Não Supervisionado

Os não supervisionados são aqueles que recebem os dados sem as *labels* (figura 8), onde devem aprender sem respostas. De acordo com (GÉRON, 2019) Alguns dos principais tipos de algoritmos não supervisionados são:

Figura 8 – Aprendizado Não Supervisionado



Fonte: (GÉRON, 2019).

a) Clustering:
   - K-Means;
   - DBSCAN;
   - Hierarchical Cluster Analysis (HCA);

b) Detecção de anomalias:
   - Isolation Forests;
   - One-class SVM;

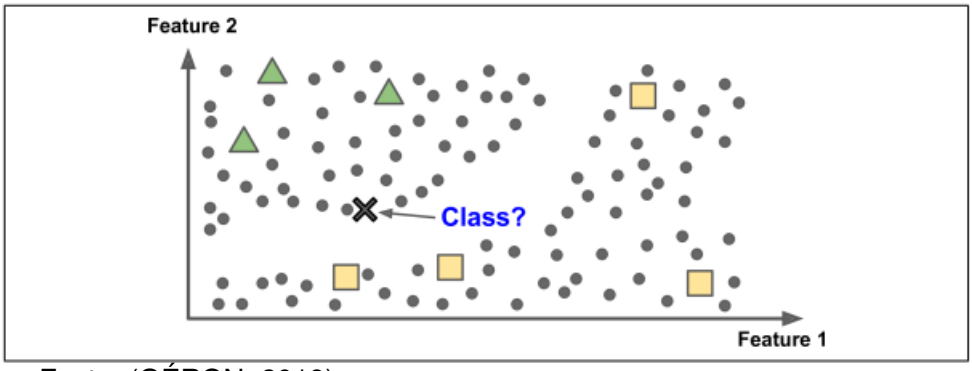
c) Redução de dimensão e visualização:
   - Principal Component Analysis (PCA);
   - Kernel PCA;
   - Local-Linear Embedding (LLE);
   - t-distributed Stochastic Neighbor Embedding (t-SNE)

d) Aprendizado pela regra de associação:
   - Apriori;
   - Eclat;

3.2.2.1.3 Semi-Supervisionado

"Alguns algoritmos conseguem lidar com conjuntos de dados de treinamento apenas parcialmente respondidos, normalmente com uma grande parte do conjunto sem *labels* e uma pequena parte com."(GÉRON, 2019). Chamado de aprendizado semi-supervisionado, normalmente consiste na combinação de dois algoritmos, um supervisionado e outro não (figura 9). Alguns dos principais são:

Figura 9 – Aprendizado Semi-Supervisionado



Fonte: (GÉRON, 2019).

a) *Self-training*: Método onde um modelo supervisionado treina em um conjunto de dados rotulados e depois é usado para rotular dados náo rotulados. As previções mais confiantes são então adicionadas ao conjunto de treinamento;

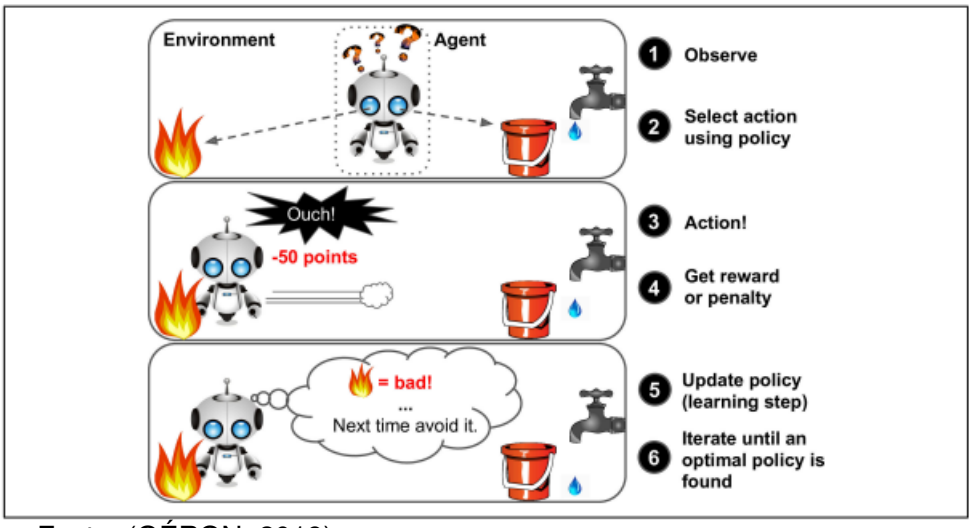
b) *Co-Training*: Dois modelos são treinados em diferentes subconjunto de características do mesmo conjunto de dados, e então cada modelo rotula dados não rotulados para o outro;

c) *Transductive Suport Vector Machines* (TSVM): Uma variação das SVMs, são usados para classificar dados não rotulados enquanto treinam em dados rotulados.

3.2.2.1.4 Reforço

O Aprendizado por reforço, diferente dos demais, consiste em um agente (sistema de aprendizado) que seleciona e realiza ações e em resposta recebe "recompensas"negativas ou positivas. Dessa forma, o agente aprende a melhor política (estratégia) para obter a maior quantidade de recompensas positivas ao longo do tempo. A política define qual ação o agente escolherá quando se encontrar em uma determinada situação (GÉRON, 2019). A figura 10 abaixo ilustra este tipo de sistema:

Figura 10 – Aprendizado por Reforço



Fonte: (GÉRON, 2019).

3.2.2.1.5 Aprendizado em lote

No aprendizado em lote, o sistema não pode adaptar seu conhecimento de forma incremental; ele requer treinamento sobre a totalidade dos dados disponíveis. Esse processo geralmente exige tempo e recursos computacionais significativos, portanto, é realizado predominantemente offline. Inicialmente, o sistema passa por treinamento e, posteriormente, é implantado na produção, onde opera sem aprendizado adicional; ele simplesmente utiliza o conhecimento adquirido (GÉRON, 2019, p. 15).

Para permitir que um sistema de aprendizado em lote reconheça novas informações (como uma nova categoria de spam), é necessário desenvolver uma nova versão do sistema do zero usando todo o conjunto de dados (incluindo os dados novos e anteriores), após o qual você deve desativar o sistema antigo e substituí-lo pela versão atualizada (GÉRON, 2019, p. 15).

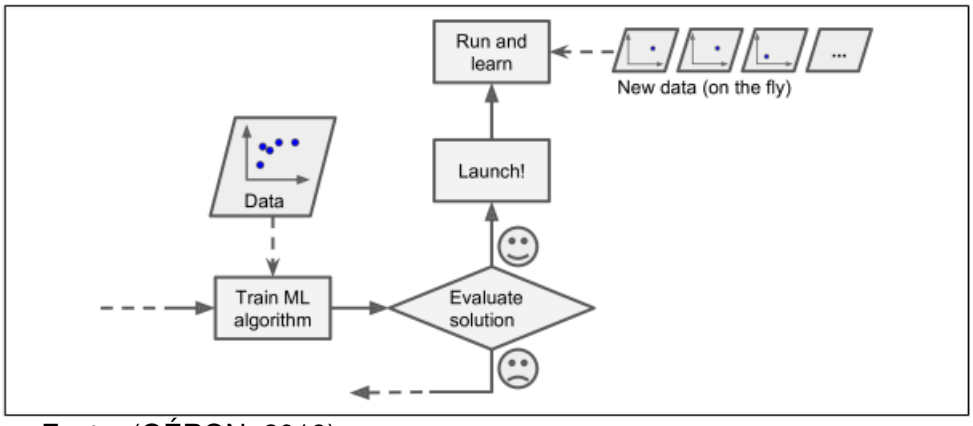
Apesar de que todo o processo de criação de um sistema de aprendizado de máquina pode ser automatizado de maneira bem fácil, fazendo com que mesmo retreinar seu sistema do zero seja uma tarefa simples e rápida, o aprendizado em lote pode trazer uma série de desafios e dificuldades; muitas vezes treinar utilizando um conjunto inteiro de dados acaba sendo muito custoso computacionalmente, fazendo-se necessária uma quantidade significativa de recursos, as vezes indisponível, ou até mesmo uma grande quantidade de tempo, de horas até dias, o que impede que ele atue em situações nas quais rápida adaptabilidade é necessário (GÉRON, 2019, p.16).

3.2.2.1.6 Aprendizado Incremental

No aprendizado online, o sistema é alimentado com pequenos "pedaços"do conjunto de dados, chamados de *mini-batches*, fazendo com que cada passo do

aprendizado seja rápido e leve possibilitando o sistema aprender na hora, conforme recebe um fluxo contínuo de dados. Como os "pedaços"já aprendidos pelo sistema não mais necessários, podem ser descartados para não ocupar espaço, o que é perfeito para sistemas online e com recursos limitados (e.g., aplicativo de celular). Na figura 11 GÉRON ilustra um exemplo de fluxo de aprendizado incremental.

Figura 11 – Aprendizado Incremental



Fonte: (GÉRON, 2019).

O parâmetro mais importante do aprendizado online em comparação com o em lote é o quão rápido o sistema deve aprender, chamado de *learning rate*. Se o sistema tiver um ritmo de aprendizado muito alto, ele irá se adaptar rápido à dados novos, mas fará com que dados antigos tenham cada vez menos peso em suas previsões, até o ponto que forem "esquecidos". Em contraste, um ritmo muito lento, aprenderá de forma mais lenta, levando muito tempo para se adaptar à novos dados.

> Um grande desafio com o aprendizado online é que, se dados ruins forem alimentados ao sistema, o desempenho do sistema diminuirá gradualmente. Se estivermos falando de um sistema em funcionamento, seus clientes vão perceber. Por exemplo, dados ruins podem vir de um sensor defeituoso em um robô, ou de alguém manipulando um motor de busca para tentar aparecer nas primeiras posições dos resultados de pesquisa. Para reduzir esse risco, você precisa monitorar seu sistema de perto e desligar rapidamente o aprendizado (e possivelmente voltar para um estado anterior que funcionava) se detectar uma queda no desempenho. Você também pode querer monitorar os dados de entrada e reagir a dados anormais (por exemplo, usando um algoritmo de detecção de anomalias). (GÉRON, 2019).

3.2.3 Floresta de Isolamento

Particularmente eficiente com alta dimensionalidade e grande quantidade de dados, utiliza uma estrutura em árvore parecida com as *decision trees* chamadas *isolation trees* (iTrees), consegue também ranquear as variáveis por importância na predição, assim ajudando a lidar com o ruído. Apesar destas características, tem dificuldade em detectar anomalias de densidade local e principalmente enclausuradas.

De acordo com FOORTHUIS (2021, p. 2) "Alguns métodos, como máquinas de vetores de suporte de uma classe ou *iForest*[2], são capazes de detectar pontos periféricos, mas não são projetados para identificar pontos enclausurados."

Segundo LIU; TING; ZHOU (2008, p. 1-2) algumas das principais características que *iForest* difere de outros modelos são:

a) Abordagem de Isolamento: diferente de métodos tradicionais que constroem perfis de instâncias normais para identificar anomalias como desvios, o *iForest* isola diretamente os pontos de dados que são anomalias. Essa abordagem inovadora não foi explorada anteriormente na literatura, segundo os autores do artigo.

b) Uso Eficiente de Subamostras: o *iForest* aproveita a subamostragem dos dados de uma maneira que não é viável em outros métodos existentes, reduzindo a necessidade de grandes conjuntos de dados de treinamento.

c) Complexidade Computacional Reduzida: o modelo tem uma complexidade de tempo linear com uma constante baixa e um baixo requisito de memória, o que o torna mais eficiente em termos de tempo de processamento e uso de recursos, em comparação com métodos baseados em distância ou densidade.

d) Eliminação da Necessidade de Perfis Normais: ao contrário de outras abordagens, o *iForest* não precisa construir um perfil normal das instâncias, simplificando o processo de detecção de anomalias e reduzindo a quantidade de dados que precisam ser processados.

e) Escalabilidade: o *iForest* é capaz de lidar com grandes conjuntos de dados e problemas de alta dimensionalidade, sendo eficiente mesmo em cenários com um grande número de variáveis irrelevantes.

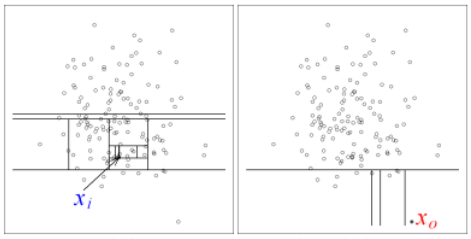
f) Eficiência com Pequeno Número de Árvores: o modelo é eficiente, exigindo apenas um pequeno número de árvores para alcançar alto desempenho na detecção de anomalias, o que o torna rápido e eficaz (LIU; TING; ZHOU, 2008, p. 1-2).

A abordagem central do *iForest* é baseada na ideia de que as anomalias são isoladas mais rapidamente em um conjunto de dados. As iTrees[3] são construídas particionando os dados recursivamente, com o objetivo de isolar instâncias individuais. Instâncias que requerem menos partições, ou seja, aquelas isoladas mais próximas à raiz da árvore, são identificadas como anomalias, devido à sua distinção dos pontos normais. A figura (12) abaixo, dos autores ilustra isso. Nela, é demonstrado como

[2] Aqui o autor se refere a *Isolation Forest*, ou Floresta de Isolamento
[3] *Isolation Trees* ou árvores de isolamento

para isolar um ponto normal nos dados necessita muitas mais partições do que um ponto anômalo, isto também explica a dificuldade do algoritmo de detectar anomalis enclausuradas ou até mesmo de densidade local.

Figura 12 – Isolamento ponto normal (xi) e ponto anômalo (x0)



(a) Isolating x_i (b) Isolating x_o

Fonte: (LIU; TING; ZHOU, 2008).

Segundo os autores: "o *iForest* pode não ser eficaz na detecção de anomalias quando estas estão fortemente sobrepostas com os dados normais"(LIU; TING; ZHOU, 2008, p. 3). Eles usam os termos "*masking*"e "*swamping*"para detalhar esta dificuldade do modelo. "*Masking* ocorre quando uma anomalia é mascarada por outros dados normais, resultando na sua não detecção."Já "*swamping* refere-se à situação em que instâncias normais são identificadas erroneamente como anomalias devido à influência de dados anômalos próximos"(LIU; TING; ZHOU, 2008, p. 2-5). Também recomendam usar um "sample-size"menor para tentar driblar estes problemas.

3.2.3.1 Parâmetros

Por ser um modelo complexo criado a partir de árvores de isolamento, que por sua vez são baseadas em árvores de decisões, possui um grande número de hiperparâmetros, sendo cada um capaz de alterar o desempenho do modelo, os hiperparâmetros são:

a) *n_estimators*: Este parâmetro especifica o número de árvores na floresta. O valor padrão geralmente é em torno de 100.

b) *max_samples*: Determina o número de amostras a serem retiradas do conjunto de dados para treinar cada estimador base. Se definido como 'auto', ele usará min(256, *n_samples*).

c) *max_features*: Especifica o número de variáveis a serem retirados do total para treinar cada estimador base.

d) *contamination*: A proporção de *outliers* no conjunto de dados. Usado ao prever anomalias para definir o limiar nas pontuações das amostras.

e) *bootstrap*: Determina se o *bootstrapping* deve ser usado para criar as amostras. Se Falso, todo o conjunto de dados é usado para construir cada árvore.

f) *n_jobs*: Indica o número de núcleos a serem usados para processamento paralelo. Defini-lo como -1 usa todos os núcleos disponíveis. O processamento paralelo pode acelerar a computação, mas também aumenta o uso de memória.

g) *random state*: Controla a geração de números pseudoaleatórios para criar as árvores de decisão. Garante a reprodutibilidade dos resultados.

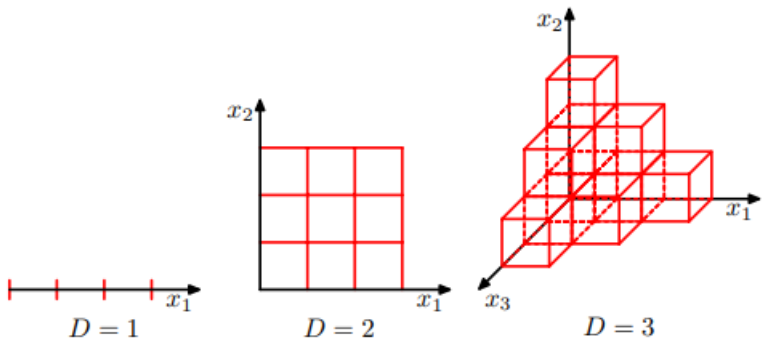
3.2.4 Maldição da Dimensionalidade

Tanto os conjuntos treino quanto os de falha possuem um problema famoso conhecido como a "maldição da dimensionalidade".

> A solução analítica efetiva de um grande número de equações, mesmo as simples, como por exemplo, equações lineares, é uma tarefa difícil. Diminuindo nossas expectativas, mesmo uma solução computacional geralmente apresenta várias dificuldades de natureza tanto grosseira quanto sutil. Consequentemente, a determinação desse máximo é definitivamente não rotineira quando o número de variáveis é grande.(BELLMAN, 1957).

Em outras palavras, em *machine learning*, quanto maior a quantidade de variáveis ou dimensões do conjunto de dados, maior o custo computacional, e a quantidade de dados de treinamento necessária para generalizar de forma precisa cresce de forma exponencial para resolver até mesmo problemas simples. (BISHOP, 2006) ilustra isso com a figura 13 abaixo.

Figura 13 – Maldição da Dimensionalidade



Fonte: (BISHOP, 2006).

Em casos de detecção de anomalia em dados de alta dimensão, segundo (ZIMEK; SCHUBERT; KRIEGEL, 2016) são enfrentados desafios como *distance concentration effect*, a presença de atributos irrelevantes e problemas de eficiência, assim como muitos outros.

O "*distance concentration effect*", podemos chamar de "o efeito de concentração de distância", é um problema onde as distâncias entre objetos independentes e

igualmente distribuídos (i.i.d.) em termos de atributos convergem para uma distribuição normal com baixa variância, levando a problemas numéricos e de parametrização.

O problema de se ter muitas dimensões é que muito atributos acabam sendo irrelevantes e podem "mascarar"os atributos relevantes, dificultando assim a detecção de *outliers* e anomalias.

Os problemas de eficiência em detectar *outliers* em conjuntos de dados com várias dimensões surgem como consequência da existência de atributos insignificantes e da necessidade de enfrentar obstáculos gerais de eficiência e eficácia.

3.2.5 Detecção de Anomalias

A detecção de anomalias é o processo de identificar ocorrências em um conjunto de dados que são incomuns e que não se ajustam aos padrões gerais observados. FOORTHUIS (2021) destaca que, apesar da longa história de pesquisas sobre o tema, as definições de anomalias continuam a ser vagas e dependem amplamente do domínio de aplicação. Além disso, a detecção de anomalias é vista como crucial tanto para a pesquisa acadêmica quanto para aplicações industriais, sendo utilizada em diversas áreas como a descoberta de fraudes, análise de qualidade de dados e controle de sistemas e processos. O mesmo observa que as anomalias podem ser tanto indesejadas, interferindo na análise de dados, quanto desejadas, fornecendo sinais importantes que os analistas procuram identificar (FOORTHUIS, 2021, p. 1-3).

BOUMAN; BUKHSH; HESKES (2023) complementam essa visão ao discutir que a detecção de anomalias, frequentemente associada à detecção de outliers no aprendizado de máquina, é uma técnica fundamental que tem sido amplamente desenvolvida ao longo dos anos para enfrentar desafios específicos, como a alta dimensionalidade dos dados. Os autores destacam que a detecção de anomalias não só é importante para identificar casos extremos, mas também para detectar padrões inusitados que podem ter implicações significativas, dependendo do contexto da aplicação (BOUMAN; BUKHSH; HESKES, 2023, p. 1).

3.2.6 Tipos de Anomalias

De acordo com BOUMAN; BUKHSH; HESKES (2023) exitem muitas definições diferentes para os tipos de anomalias, porém muitas falham em encapsular um amplo espectro de propriedades ou categorias de anomalias. Sendo assim, os autores tratam as anomalias como capazer de possuir múltiplas propriedades:

> Definimos quatro escalas de propriedades não exclusivas que, quando combinadas, abrangem todos os tipos de anomalias encontradas em dados tabulares multivariados conhecidos na literatura que nos é familiar. (BOUMAN; BUKHSH; HESKES, 2023, p.4).

Essas quatro escalas são:

a) Anomalias Enclausuradas e Periféricas:
   - Anomalias Enclausuradas: essas anomalias estão cercadas por dados normais no espaço de características, o que as torna difíceis de detectar, pois estão embutidas em regiões densas de dados normais;
   - Anomalias Periféricas: essas anomalias ocupam as bordas do espaço de características e têm uma ou mais pontuações de atributos abaixo do mínimo ou acima do máximo das regiões normais, tornando-as mais evidentes para detecção.

b) Anomalias de Densidade Global e Local:
   - Anomalias Globais: são pontos que podem ser isolados dos dados normais porque ocupam uma região de baixa densidade no espaço de características;
   - Anomalias Locais: estas anomalias estão localizadas em regiões onde a densidade é baixa em comparação com as regiões normais próximas. Isso permite que anomalias bem definidas sejam identificadas mesmo em conjuntos de dados com múltiplos clusters de diferentes densidades.

c) Anomalias Isoladas e Agrupadas:
   - Anomalias Isoladas: geralmente consistem em pontos de dados únicos, sem outros pontos anômalos ou normais próximos, sendo mais fáceis de identificar devido à sua singularidade;
   - Anomalias Agrupadas: em alguns casos, pequenas aglomerações de anomalias formam clusters, levando a "anomalias agrupadas", onde anomalias semelhantes podem mascarar a presença uma da outra ao formarem um cluster.

d) Anomalias Univariadas e Multivariadas:
   - Anomalias Univariadas: podem ser identificadas usando apenas uma característica, onde um único valor fora do intervalo normal indica a anomalia;
   - Anomalias Multivariadas: necessitam de uma combinação específica de valores de características para serem identificadas como anômalas, muitas vezes relacionadas a desvios na estrutura de dependência normal dos dados.

4 DEFINIÇÃO DO PROBLEMA

O problema abordado neste trabalho é o de detectar anomalias advindas de uma família de falhas a partir de dados coletados e disponibilizados em séries temporais de um sistema fotovoltaico em laboratório, utilizando análise de dados e aprendizado de máquina. O problema inclui: Análise das falhas, pré-processamento dos dados, criação de um ou mais modelos de aprendizado de máquina, parametrização, criação de um método avaliativo, otimização das etapas de aprendizagem e, por fim, seleção do melhor modelo com base no método avaliativo desenvolvido anteriormente.

O trabalho delimita-se apenas na geração e avaliação do modelo utilizando algoritmos existentes. O modelo será capaz de identificar anomalias, e apresentar onde elas estão, que levem a uma possível falha com antecedência, porém não será capaz de identificar qual falha é responsável pelas mesmas. As medições com e sem falhas provém de um conjunto de dados de aproximadamente 15 segundos, onde as anomalias começam a aparecer por volta de 7 segundos.

Essa pesquisa não irá abranger a realização de medições de uma instalação fotovoltaica, desenvolvimento de uma solução comercial, muito menos um instrumento de identificação de uma ou mais falhas, servindo apenas como um indicador de anomalias detectadas. Portanto, fica evidenciado que a pesquisa tem como objetivo o desenvolvimento de um modelo de aprendizado de máquina que seja capaz de identificar anomalias nos dados entre medições em condições estáveis e medições precedentes a uma família de falhas, utilizando análise dos valores de mais de 100.000 medições de um total de 14 segundos de experimento e 14 categorias.

Para o desenvolvimento de um modelo funcional (que consiga detectar anomalias reais acusando o mínimo de falsos positivos), alguns fatores são necessários: coletar e armazenar dados de uma instalação fotovoltaica (ou de uma simulação minimamente precisa); filtragem adequada dos dados, para garantir uma perda mínima de informações; treinamento de modelos em dados de funcionamento normal (onde não há a ocorrência de anomalias, para que os modelos entendam que aquele é o “padrão” desejado); aplicação dos modelos treinados em dados onde há falha; avaliação do desempenho dos modelos e, por fim, seleção do modelo de melhor desempenho.

Visando o objetivo proposto, será desenvolvido um sistema que será alimentado com as medições de uma instalação fotovoltaica funcionando de duas maneiras diferentes controladas por um sistema baseado na técnica de *particle swarm optimization* (PSO) para garantir o Rastreamento do Ponto Máximo de Potência *Maximum Power Point Tracker* (MPPT) quando o nível de potência disponível é inferior ao nominal, e o Rastreamento do Ponto Intermediário de Potência *Intermetiate Power Point Tacker* (IPPT) quando o nível de potência disponível é superior ao nominal, e em ambas o

sistema simula o estado normal e o estado precedente a 7 falhas diferentes. Cada medição contará com as informações: Tempo, Ipv, Vpv, Vdc, ia, ib, ic, va, vb, vc, Iabc, If, Vabc, Vf.

4.1 Etapas de Atividades

As atividades deste projeto foram organizadas em quatro fases: concepção, elaboração, construção e transição:

a) Concepção: a fase de concepção é a primeira etapa do processo de desenvolvimento. Seu objetivo principal é estabelecer a visão do projeto e determinar sua viabilidade. Durante esta fase, são realizadas atividades como:
   - Análise da viabilidade técnica;
   - Identificação dos principais requisitos.

b) Elaboração: a fase de elaboração foca em refinar os requisitos e planejar a arquitetura do sistema. As atividades principais incluem:
   - Desenvolvimento da arquitetura: definir a arquitetura do sistema, incluindo a estrutura e os componentes principais;
   - Avaliação de modelagem: verifica a viabilidade das atividades anteriores.

c) Construção: a fase de construção é onde o sistema é realmente desenvolvido. As atividades incluem:
   - Refinamento dos requisitos: detalhar e priorizar os requisitos coletados na fase de concepção;
   - Desenvolvimento e implementação: codificar e integrar os componentes do sistema.

d) Transição: a fase de transição envolve a implantação final e validação dos resultados:
   - Testes: realizar testes para garantir a qualidade do projeto;
   - Validação dos resultados: definir e explicar se os resultados atingem o objetivo principal;
   - Pós-validação: definir de possíveis mudanças e/ou estudos futuros.

5 PROPOSTA DE SOLUÇÃO

Tendo em vista a necessidade de funcionamento constante em usinas fotovoltaicas, a detecção de anomalias pode ser uma ótima ferramenta para evitar danos posteriores ou até mesmo a operação com componentes danificados, para isso, o modelo deverá ser rápido e sensível à *outliers*, pois no caso da detecção de anomalias advindas de uma possível falha, falsos negativos são muito mais prejudiciais que falsos positivos.

Para solucionar o problema, este trabalho propõe a utilização de alguns modelos de aprendizagem de máquina e a utilização de uma métrica existente para avaliar contextualmente o problema e comparar os resultados dos modelos. Para isso os seguintes passos serão executados:

a) Entendimento do conjunto de dados;
b) Seleção dos modelos de machine learning;
c) Tratamento dos dados;
d) Aplicação dos modelos nos dados;
e) Avaliação dos resultados;
f) Seleção do melhor modelo.

5.1 Entendendo os Dados

Os dados utilizados neste projeto foram baseados no estudo de BAKDI *et al.* (2021), onde um sistema fotovoltaico (PV) conectado à rede foi implementado em laboratório para a coleta de informações relevantes. Neste contexto, "para validar o desempenho de métodos baseados em dados contra falhas reais sob modos MPPT/IPPT e condições práticas"(BAKDI *et al.*, 2021, p. 4). O sistema foi equipado com uma unidade de medição de fasores e diversos sensores de alta frequência, permitindo a aquisição de sinais em tempo real, como "a tensão e corrente do *array* PV, a tensão DC, e as correntes trifásicas da rede"com um tempo de amostragem de 100 microssegundos (BAKDI *et al.*, 2021, p. 4). As falhas foram injetadas manualmente em diferentes componentes, como o inversor e os controladores MPPT/IPPT, fornecendo um conjunto robusto de dados que foi utilizado neste projeto para testar novos métodos de detecção de falhas sob condições reais de operação (BAKDI *et al.*, 2021, p. 5).

5.1.1 Falhas

O autor categoriza as falhas como:

a) Falha no Inversor (F1): É descrita como uma "falha completa em um dos seis IGBTs"do inversor (BAKDI *et al.*, 2021, p. 7);

b) Falha no Sensor de Feedback de Corrente (F2): Esta falha envolve um "defeito em um dos sensores de corrente, com 20% de erro"(BAKDI *et al.*, 2021, p. 7);

c) Anomalia na Rede (F3): Refere-se a "quedas de tensão intermitentes"(BAKDI *et al.*, 2021, p. 7);

d) Desajustes na Matriz PV (F4 e F5): O autor descreve dois tipos de desajustes: o F4, que envolve "sombreamento parcial não homogêneo entre 10% e 20%", e o F5, que é um "circuito aberto de 15% na matriz PV"(BAKDI *et al.*, 2021, p. 7);

e) Falhas no Controlador MPPT/IPPT (F6 e F7): O F6 é uma "redução de 20% no ganho do controlador PI do MPPT/IPPT", enquanto o F7 refere-se a um "aumento de 20% na constante de tempo do controlador PI"(BAKDI *et al.*, 2021, p. 7).

O autor também relata algumas características das falhas:

a) F1 e F3: "As falhas F1 e F3 que ocorrem no lado da rede do sistema PV conectado à rede são fáceis de detectar, pois afetam apenas o lado AC, onde os dados apresentam variabilidade muito pequena"(BAKDI *et al.*, 2021, p. 5-6).

b) F2: "Esta falha é mais severa, comparada à F6 acima, pois gera um erro de estado estacionário diferente de zero e leva a uma configuração incorreta dos controladores de feedback"(BAKDI *et al.*, 2021, p. 11-12).

c) F4 e F5: "Os desajustes na matriz PV, como F4 e F5, são desafiadores de detectar devido à grande variabilidade nos dados dos sensores no lado DC; felizmente, essas falhas são de níveis de gravidade mais baixos, causando apenas perdas de energia"(BAKDI *et al.*, 2021, p. 5-6).

d) F6 e F7: "Essas falhas são amplamente comuns na prática, seu impacto nos sistemas PV conectados à rede, descrição teórica e características I-V são bem detalhadas em uma revisão abrangente em "*A comprehensive review on protection challenges and fault diagnosis in PV systems*"dos autores (Dhanup S. Pillai, N. Rajasekar, 2018)"(BAKDI *et al.*, 2021, p. 5-6)

5.1.2 Conjunto de Dados

Ao carregar os conjuntos de dados e observar as 5 primeiras linhas, com o código: "*.head*()", é constatado com a figura 14 que: de fato as medições ocorrem a cada 100us; o conjunto possui 13 variáveis (dimensões), 14 se contar a variável

"Tempo"(*Time*) sendo usada como *Index*; os valores de cada variável estão em grandezas diferentes; e alguns valores estão negativos:

Figura 14 – HEAD

Time	Ipv	Vpv	Vdc	ia	ib	ic	va	vb	vc	Iabc	If	Vabc	Vf
0.000028	1.572327	101.348877	144.140625	-0.135133	0.490112	-0.354985	41.744537	-149.872894	109.064585	1.0	50.0	1.0	50.0
0.000128	1.503265	101.458740	143.554688	-0.108277	0.510254	-0.388555	46.831512	-150.716705	105.829976	1.0	50.0	1.0	50.0
0.000228	1.492859	101.574707	143.554688	-0.168702	0.496826	-0.334844	51.074677	-152.018585	102.543132	1.0	50.0	1.0	50.0
0.000328	1.558136	101.312256	143.261719	-0.135133	0.510254	-0.361699	55.848236	-152.585144	98.143260	1.0	50.0	1.0	50.0
0.000428	1.631927	101.141357	143.847656	-0.202271	0.503540	-0.321416	60.055237	-152.609253	94.261729	1.0	50.0	1.0	50.0

Fonte: (Autor, 2023).

Observando as informações do conjunto com ".*info*()", figura 15, é observado que não há valores nulos ou faltando e que todos os valores são do tipo "*float*64", significando que todos são valores numéricos reais com casa decimais, ou seja, todos quantitativos:

Figura 15 – INFO

```
<class 'pandas.core.frame.DataFrame'>
Float64Index: 143715 entries, 2.81780014539379e-05 to 14.3699749274729
Data columns (total 13 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Ipv     143715 non-null  float64
 1   Vpv     143715 non-null  float64
 2   Vdc     143715 non-null  float64
 3   ia      143715 non-null  float64
 4   ib      143715 non-null  float64
 5   ic      143715 non-null  float64
 6   va      143715 non-null  float64
 7   vb      143715 non-null  float64
 8   vc      143715 non-null  float64
 9   Iabc    143715 non-null  float64
 10  If      143715 non-null  float64
 11  Vabc    143715 non-null  float64
 12  Vf      143715 non-null  float64
dtypes: float64(13)
memory usage: 15.4 MB
```

Fonte: (Autor, 2023).

De acordo com BAKDI *et al.* (2021), as variáveis são:

a) Ipv: Corrente de saída do sistema fotovoltaico.

b) Vpv: Tensão de saída do sistema fotovoltaico.

c) Vdc: Tensão em corrente contínua (DC).

d) ia, ib, ic: Correntes nas fases a, b e c, respectivamente. Essas variáveis representam as correntes trifásicas no sistema, correspondendo a cada uma das três fases da alimentação elétrica.

e) va, vb, vc: Tensões nas fases a, b e c, respectivamente. Estas variáveis medem as tensões nas três fases do sistema.

f) Iabc: Indicador binário, mostrando o estado da corrente trifásica (se está ou não ativo).

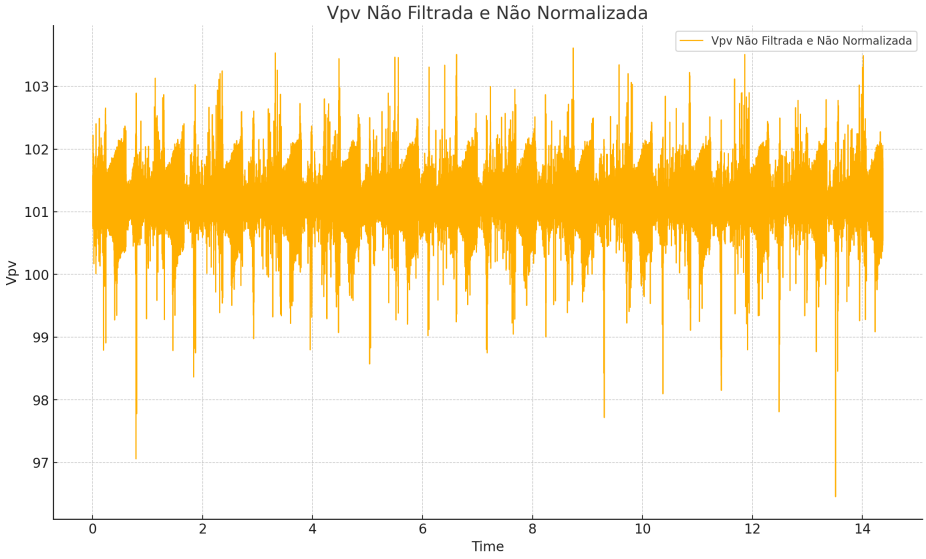
g) If: Frequência da corrente.

h) Vabc: Indicador binário, mostrando o estado da tensão trifásica (se está ou não ativo).

i) Vf: Frequência da tensão.

Sabendo o que é cada variável, pode-se agora "plotar"gráficos para analisar melhor o comportamento das mesmas. Abaixo, figura 16, o gráfico da tensão de saída do sistema fotovoltaico sem falhas, não filtrada:

Figura 16 – Vpv sem falhas não filtrada



Fonte: (Autor, 2024).

A grande quantidade de ruídos impede qualquer tipo de análise e conclusão nos gráficos, por isso será usado para análise apenas gráficos filtrados e feitos a partir dos conjuntos de falhas, onde podemos observar as anomalias. Abaixo, figura 17, uma comparação dos dados filtrados e não filtrados do gráfico acima:

Figura 17 – Vpv sem falhas, filtrado e não filtrado

figuras/Vpv0Lfiltrado.png

Fonte: (Autor, 2024).

Como descrito no capítulo anterior, os conjuntos de dados que possuírem potência disponível maior que a nominal, serão nomeados com a letra "L"de limitada, e antecedendo a letra o número correspondente com a sua falha, 0 sendo sem falhas e de 1 a 7 com as falhas, descritas neste capítulo, em ordem. Analogamente, os conjuntos com potência dispopnível menor que a nominal possuirão a letra "M"e seguindo a mesma lógica anterior.

5.1.3 Analisando os Gráficos

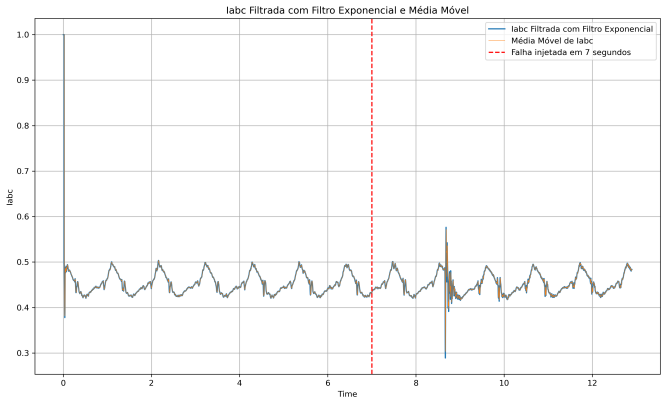
Neste capítulos serão analisadas e classificadas as anomalias de cada conjunto de dados, as variáveis "Vpv", "Ipv", "Vdc", "Vabc"e "Iabc"caso necessário as frequências também serão analisadas. Para manter o capítulo mais organizado, apenas um gráfico de cada falha para cada tipo será apresentado.

5.1.3.1 Falhas em Potência Limitada

Começando pelas falhas em potência limitada, abaixo segue o gráfico e a análise de cada falha:

a) Falha 1: de acordo com BAKDI *et al.* (2021) esta falha é de fácil detecção e afeta apenas o lado da corrente alternada, com pouca variabilidade. Observando o gráfico da corrente trifásica abaixo, figura 18, é possível ver as anomalias presentes por volta dos 9 segundos. De acordo com as descrições de BOUMAN; BUKHSH; HESKES 2023 estas anomalias são do tipo global com as características periférica e possivelmente agrupada. Justificativa: A mudança brusca no comportamendo da variável, causando um desvio significativo do padrão anterior indica uma variável global. É periférica pois ela se destaca do restante dos dados, ocorrendo na borda do comportamento normal esperado. Se outras variáveis apresentarem um comportamento semelhante, ela pode ser considerada parte de um grupo, um "*cluster*".

Figura 18 – Iabc com falha 1

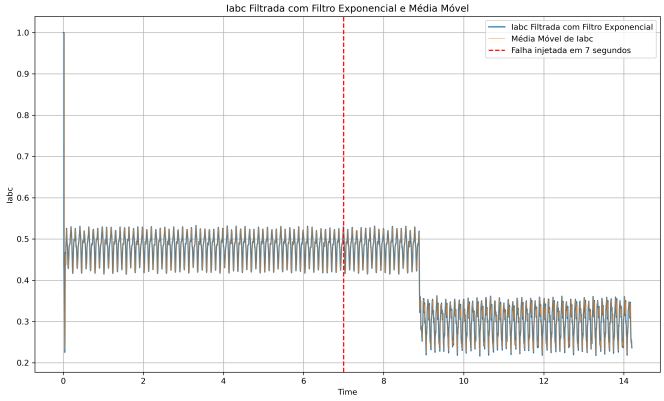


Fonte: (Autor, 2024).

b) Falha 2: de acordo com o autor esta falha é mais grave, então as anomalias devem ser detectadas o mais rápido possível. O gráfico da corrente trifásica abaixo, figura 19, nos mostras que, por volta dos 9 segundos, a falha muda completamente a tendência da corrente, indo além do mínimo

estabelecido anteriormente, por volta de 0,4A. Então podemos classificar como global e periférica, pelos mesmos motivos da falha anterior, porém, esta aparenta possuir a característica de isolada, por estabelecer um novo padrão e não retornar ao anterior.

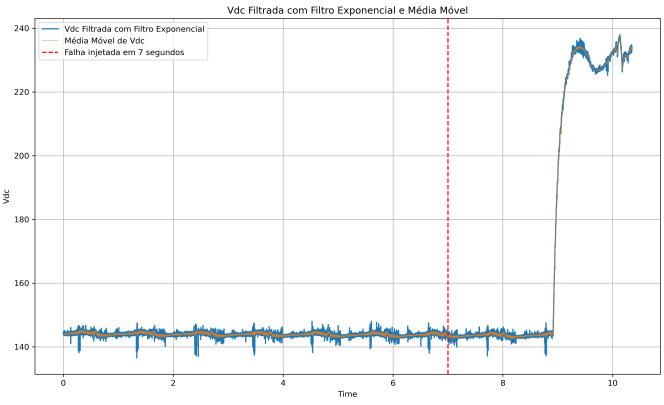
Figura 19 – Iabc com falha 2



Fonte: (Autor, 2024).

c) Falha 3: falha também de detecção fácil, na figura 20 abaixo, as anomalias são do tipo global, com as características periféricas e isoladas, justificado anteriormente.

Figura 20 – Vdc com falha 3

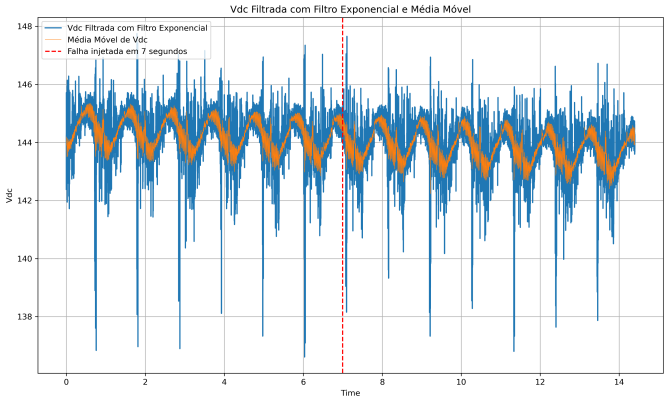


Fonte: (Autor, 2024).

d) Falha 4: falha de gravidade baixa e desafiadora detecção. Devido a variabilidade de dados no lado DC a detecção é muito difícil, observado na figura 21 abaixo, as anomalias são classificadas como do tipo local, com as características enclausuradas e multivariadas. Justificativa: após os 7 segundos, não há uma mudança drástica no comportamento geral, mas

podemos observar pequenas variações que se destacam das flutuações normais anteriores. Essas mudanças representam pequenas variações na densidade local em comparação com o comportamento anterior. Também parecem estar dentro do padrão geral do sinal, isso sugere que são pequenas variações dentro da estrutura normal dos dados. Se considerarmos que estas pequenas variações estão relacionadas com outras variáveis, ou são resultado dependência entre múltiplas variáveis, então podem ser multivariadas.

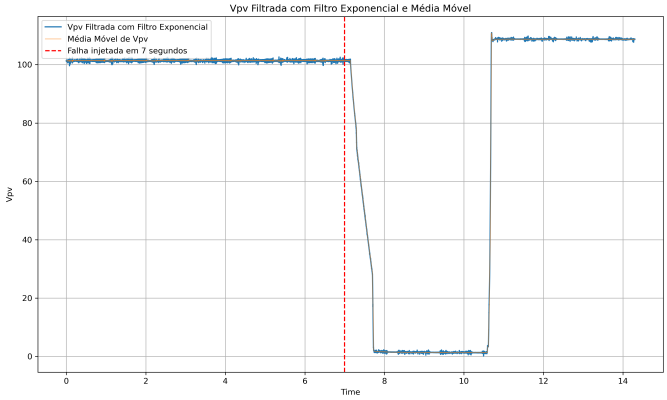
Figura 21 – Vdc com falha 4



Fonte: (Autor, 2024).

e) Falha 5: falha com a mesma descrição pelo autor, porém com detecção visível em diversas variáveis. As anomalias observadas na figura 22 abaixo, são do tipo global, periféricas e isoladas.

Figura 22 – Vpv com falha 5

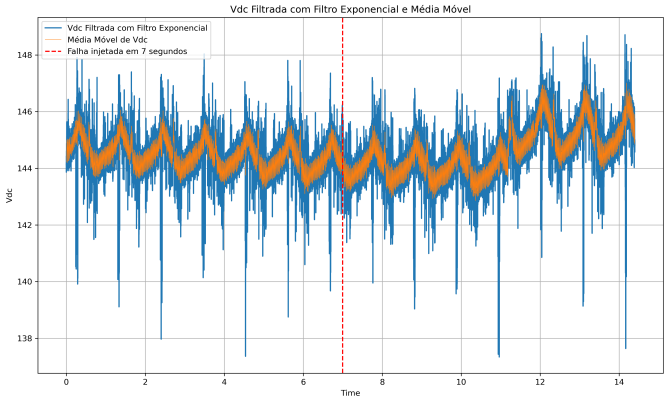


Fonte: (Autor, 2024).

f) Falha 6: Tanto esta quanto a falha 7 são consideradas pelo autor comuns

na prática, no caso da falha 6, figura 23 abaixo, as anomalias podem ser categorizadas como do tipo local, enclausuradas e multivariadas.

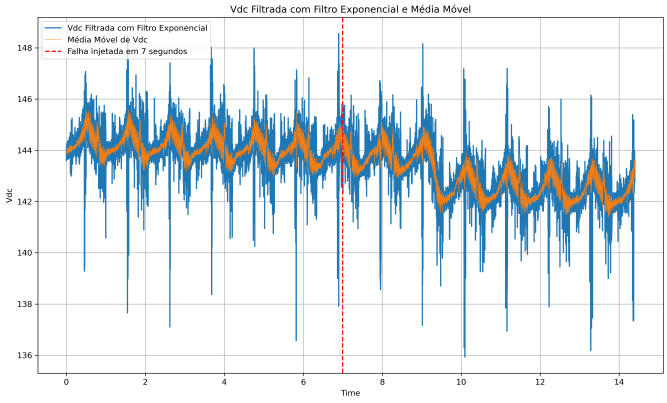
Figura 23 – Vdc com falha 6



Fonte: (Autor, 2024).

g) Falha 7: Também na tensão direta (Vdc) as anomalias no gráfico, figura 24, abaixo podem ser categorizadas como local, enclausuradas e multivariadas

Figura 24 – Vdc com falha 7

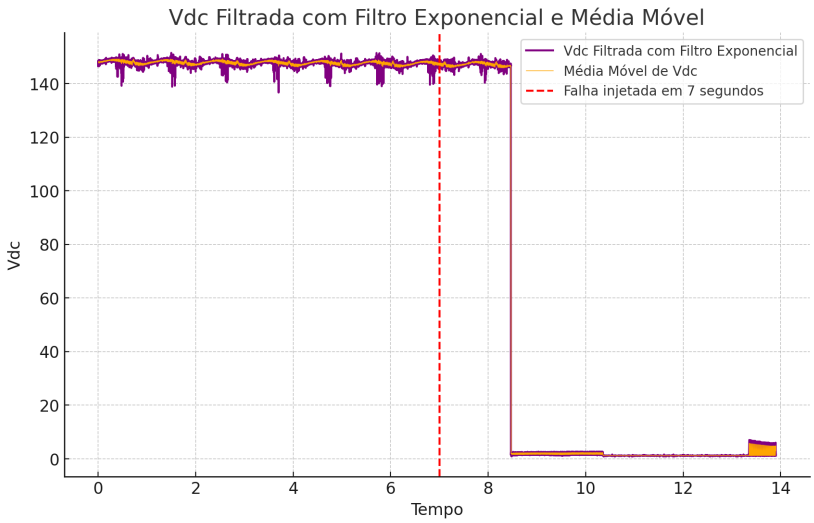


Fonte: (Autor, 2024).

5.1.3.2 Falhas de Pontência Máxima

a) Falha 1:Há uma queda acentuada no valor de Vdc, o que claramente indica uma anomalia. Essa queda poderia ser considerada uma anomalia do tipo global, caraterizada como periférica e isolada. Observado na figura 25 abaixo:

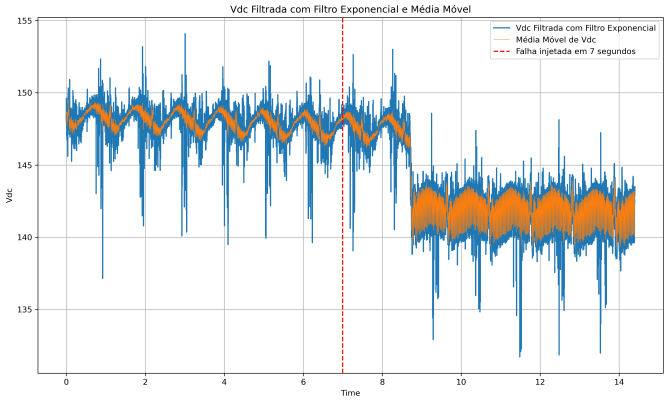
Figura 25 – Vdc com falha 1



Fonte: (Autor, 2024).

b) Falha 2: Anomalias do tipo global, caracterizadas como periféricas e isoladas. Visto na figura 26 abaixo:

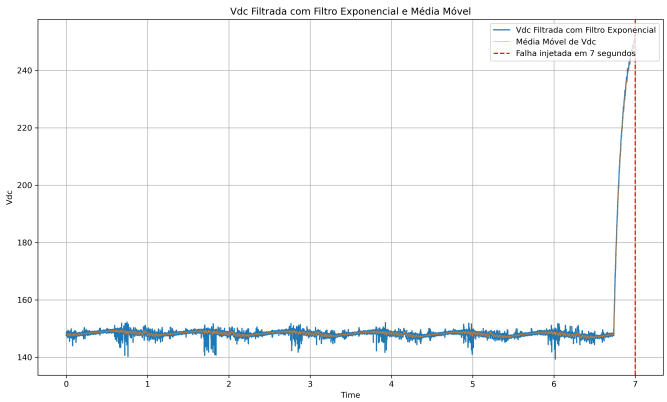
Figura 26 – Vdc com falha 2



Fonte: (Autor, 2024).

c) Falha 3: global, periférica e isolada, observado na figura 27 abaixo:

Figura 27 – Vdc com falha 3



Fonte: (Autor, 2024).

d) Falha 4, 6 e 7: Possuem um comportamento no gráfico muito semelhante à seus pares de potência limitada, levando assim há possuírem anomalias com as mesmas classificações e características: local, enclausuradas e multivariadas.

e) Falha 5: Possuindo um gráfico semelhante com seu par de potência limitada, suas anomalias levam as mesmas características, global, periféricas e isoladas.

5.2 Seleção do Modelo

Com as características e desafios do conjunto de dados em mente, o modelo selecionado deverá:

a) Lidar com a alta dimensionalidade do conjunto;

b) Lidar com uma certa quantidade de ruído, seja de varíaveis irrelevantes ou de medições ruidosas;

c) Conseguir identificar uma mudança súbita nos dados em alguns conjuntos e mudanças quase imperceptíveis em outros.

Tendo isso em mente, o seguinte modelo foi selecionado: Floresta de Isolamento (*Isolation Forest*)

De acordo com os autores, *iForest* pode ser utilizado de duas formas para detectar anomalias. A primeira é treinando o algoritmo com um conjunto de dados com *labels*[1], chamado de conjunto de treinamento, em seguida realizar as predições em um conjunto sem *labels*, ou seja, para testes. A segunda é executando o treinamento em um conjunto de treino sem respostas, neste caso não há a necessidade de *labels*. Pelo

[1] Chamadas desta forma, são as respostas que o algoritmo quer encontrar com sua predição, no caso de anomalias, seria uma variável dizendo se é anômalo ou não

fato de que os conjuntos de dados sendo utilizados neste trabalho não possuem *labels*, apenas a última será utilizada. Segundo os autores não há diferença significativa de performance.

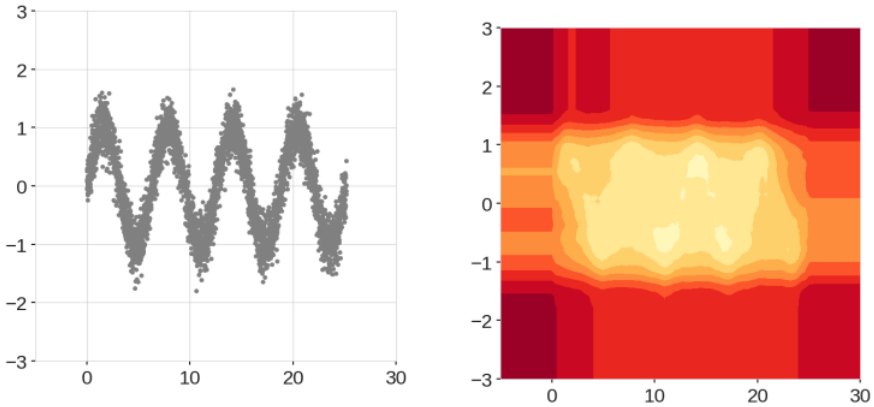
5.2.1 Floresta de Isolamento Extendida

Segundo os autores de HARIRI; KIND; BRUNNER (2021) há defeitos no algoritmo padrão do *iForest* que precisam ser adereçados antes de usar seus "*scores*"no processamento de dados. Segundos os mesmos, essa inconsistência nos *scores* é causada pelas divisões dos dados que são sempre paralelas aos eixos. "Os mapas de escores de anomalia gerados pelo Isolation Forest padrão sofrem de artefatos introduzidos pelo critério de ramificação da árvore binária"(HARIRI; KIND; BRUNNER, 2021, p. 1). Esses artefatos levam a falhas na identificação correta de anomalias, comprometendo a precisão do modelo.

Para superar essas deficiências, os autores propuseram o *Extended Isolation Forest* (EIF), que introduz cortes baseados em hiperplanos com inclinações aleatórias. Essa abordagem corrige o viés observado no *iForest* original e melhora significativamente a consistência dos *scores* de anomalia. Como afirmado pelos autores, "a EIF elimina os artefatos que levam a falsas detecções e omissões, permitindo uma análise mais precisa"(HARIRI; KIND; BRUNNER, 2021, p. 6). Com isso, o EIF consegue lidar melhor com anomalias locais, preservando ao mesmo tempo a eficiência computacional do modelo original.

A imagem abaixo, figura 28, demonstra o viés. Nela é usado um *heatmap*, (a) para demonstrar os *scores* de anomalia gerados pelo *iForest* quando aplicado aos dados sinusoidoias, (b) ao lado, onde quanto mais claro, maior o *score*.

Figura 28 – *Heatmap* da predição do *iForest* demonstrando o viés



(a) Sinusoidal data points with Gaussian noise.

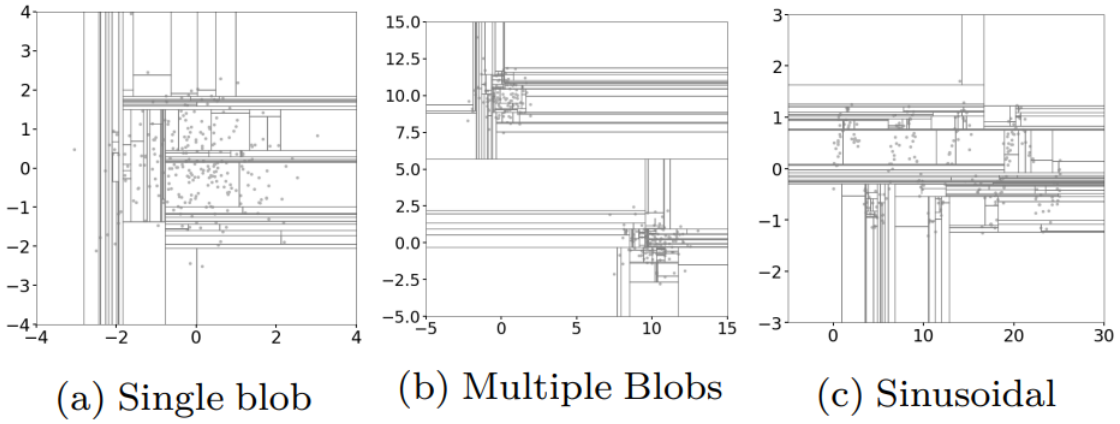
(b) Anomaly Score Map

Fonte: (HARIRI; KIND; BRUNNER, 2021, p. 3).

A imagem 29 abaixo demonstra uma *iTree* particiona os dados para indetifi-

car anomalias. A primeira (a), dados em formato circular com ruído, a segunda (b) dois circulos com ruído, e por último os dados em formato sinusoidal com ruído.

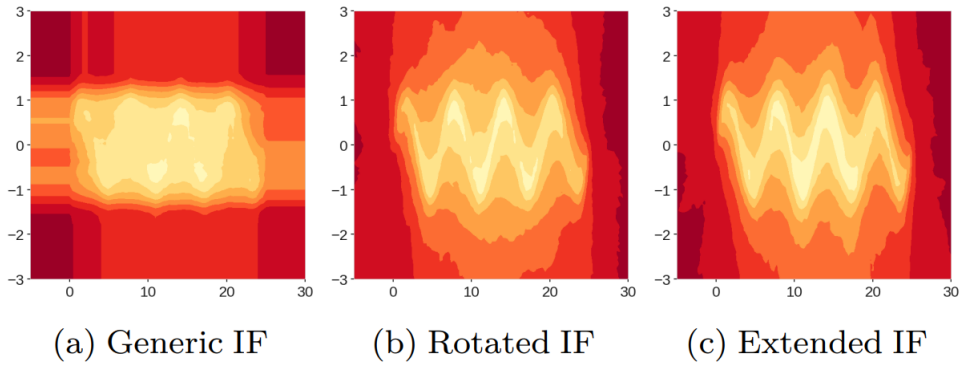
Figura 29 – Dados particionados por uma *iTree*



Fonte: (HARIRI; KIND; BRUNNER, 2021, p. 4).

E por último, a imagem 30 abaixo compara os *heatmaps* dos *scores* do *iForest* padrão, para uma versão rotacionada e para a versão extendida no conjunto de dados sinusoidais.

Figura 30 – Comparação dos *heatmaps* dos algoritmos



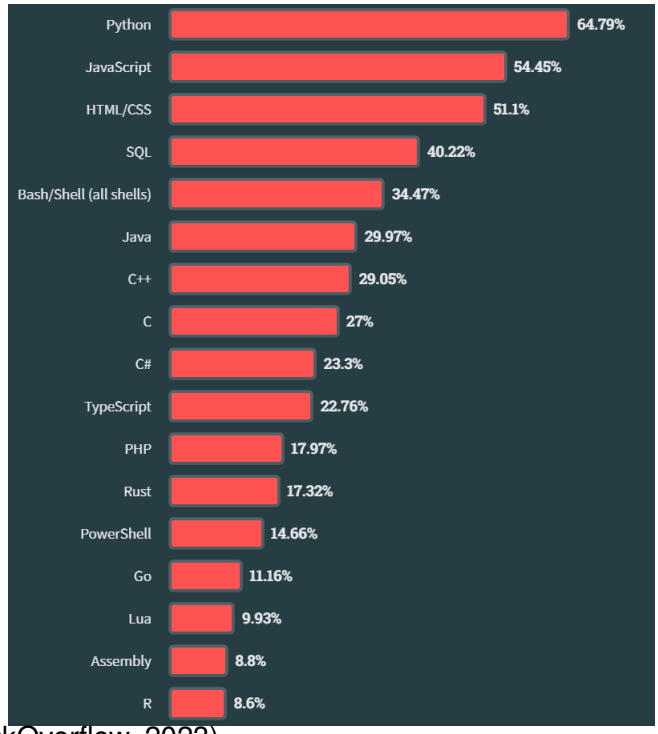
Fonte: (HARIRI; KIND; BRUNNER, 2021, p. 8).

5.3 Tecnologias Utilizadas

Nas áreas de análise e ciência de dados, algumas tecnologias são imprescindíveis para garantir o sucesso e a fluidez de um projeto bem executado. Algumas se tornaram referência na área, facilitando e expandindo as fronteiras do que é possível e acessível de se realizar com dados.

Sendo *Python*, R e SQL os duas principais linguagens da área, *Python* se tornou a terceira linguagem de programção mais utilizada hoje em dia, de acordo com a pesquisa anual do *StackOverflow* porém fica em primeiro em realação aos

Figura 31 – Linguagens Mais Usadas por não Desenvolvedores



Fonte: (StackOverflow, 2023).

não desenvolvedores, não profissionais ou aos que estão aprendendo, figura (31), graças a sua flexibilidade, simplicidade e legibilidade, o que a torna fácil de aprender. Outras tecnologias utilizadas neste projeto também foram abordadas na pesquisa, como *pandas*, *numpy* e *scikit-learn*.

Pandas criada em 2011 por (MCKINNEY, 2011) se tornou a base e a fundação da acessibilidade e sucesso da ciência de dados hoje em dia, sendo segundo *StackOverflow*, a terceira biblioteca/*framework* mais utilizada. *Numpy*, a segunda biblioteca mais popular, criada por (WALT; COLBERT; VAROQUAUX, 2011) possibilita a utilização e operações com *arrays*, matrizes de dados e funções matemáticas de alto nível. *Scipy* é uma biblioteca grátis e *open-source* usada para computação científica e técnica, possibilita a utilização de algebra linear, interpolação, otimização, transformada discreta de Fourier, entre outros. Matplotlib é uma biblioteca criada para construir gráficos utilizando *Python* e *Numpy*, foi criada por John D. Hunter em 2003 que a descreve da seguinte forma em seu texto introdutório: "Matplotlib é uma biblioteca para fazer gráficos 2D de *arrays* em *Python*. Embora tenha suas origens na emulação dos comandos gráficos do MATLAB, é independente do MATLAB e pode ser usada de uma maneira "*Pythonic*"e orientada a objetos. Embora o Matplotlib seja escrito principalmente em *Python* puro, ele faz uso intenso de *NumPy* e outro código de extensão para fornecer bom desempenho, mesmo para *arrays* grandes".

5.4 Primeiras Predições

Para iniciar o desenvolvimento, é utilizado tanto o *iForest* quanto o EIF nos dados antes de qualquer modificação ou tratamento, assim é possível observar a versatilidade dos algoritmos junto com suas limitações.

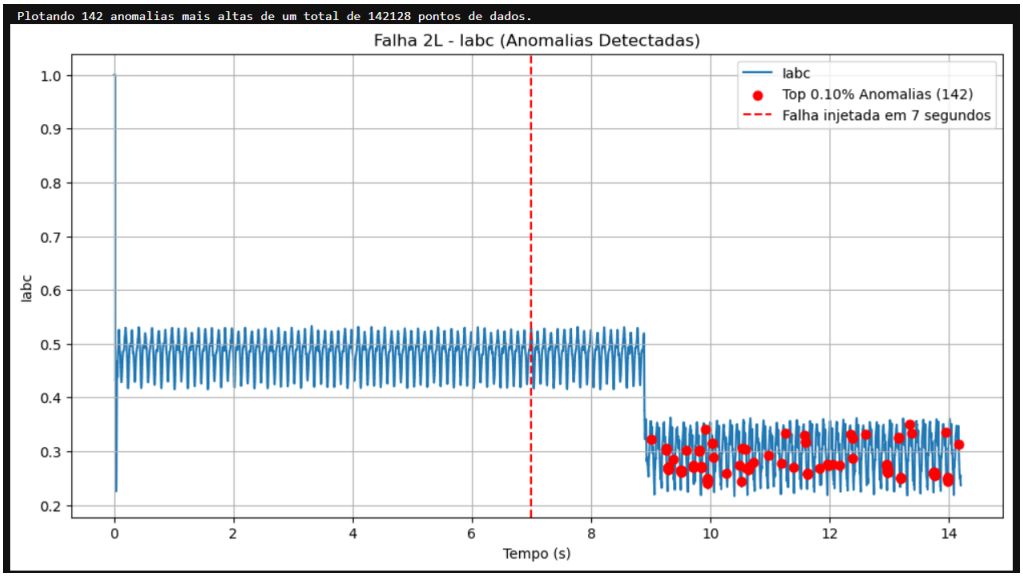
Como os conjuntos de dados tem por volta de 140 mil instâncias, será adicionado no gráfico apenas 0,1% dos pontos com "*anomaly_score*"[2], esses pontos serão considerados como anomalia e marcados como pontos vermelhos no gráfico.

Irá ser observado o desempenho da versão padrão e da versão extendida do algoritmo nos diferentes tipos de anomalias, e nas diferentes potências de dados.

5.4.1 Floresta de Isolamento

Como descrito anteriormente, o algoritmo conseguiu bons resultados praticamente todos os conjuntos de dados com falhas de densidade globas, principalmente com anomalias isoladas e periféricas, tanto de potência limitada quanto de potência máxima, ilustrado nas figuras 32 e 33 respectivamente.

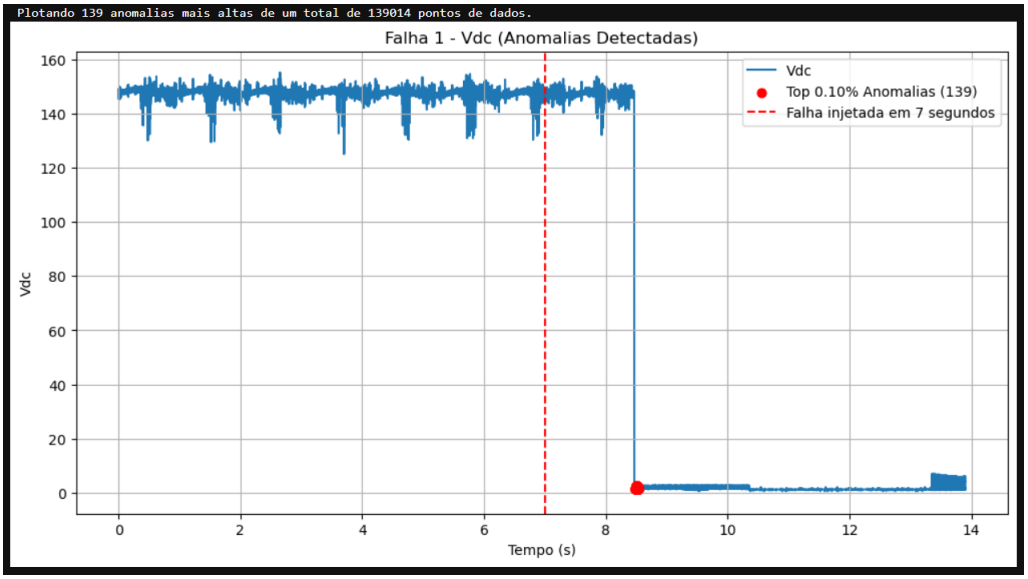
Figura 32 – Top 0,1% de Anomalias Falha 2L



Fonte: (do Autor, 2024).

[2] Grau de certeza do algoritmo em relação a um ponto ser anomalia

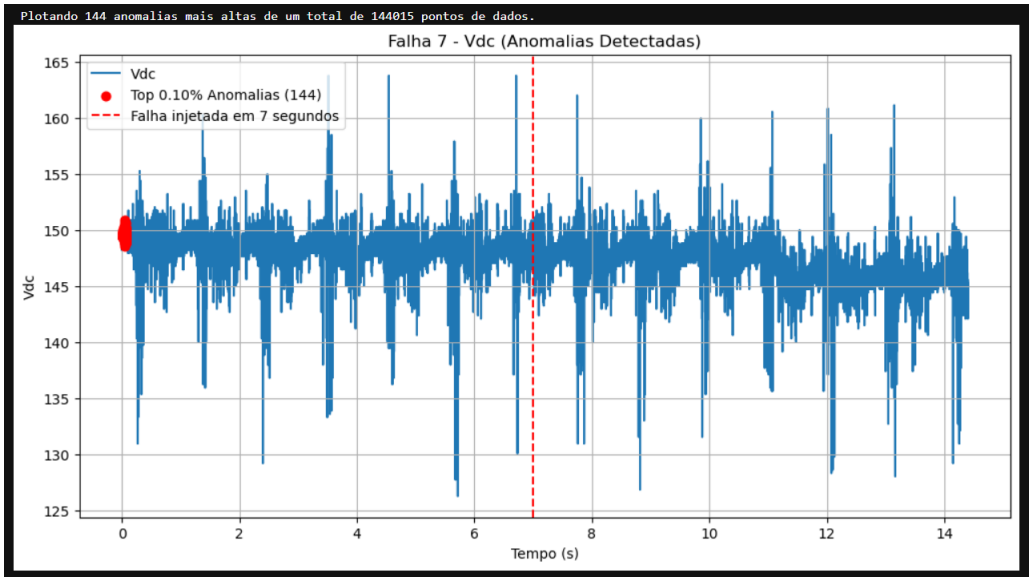
Figura 33 – Top 0,1% de Anomalias Falha 1M



Fonte: (do Autor, 2024).

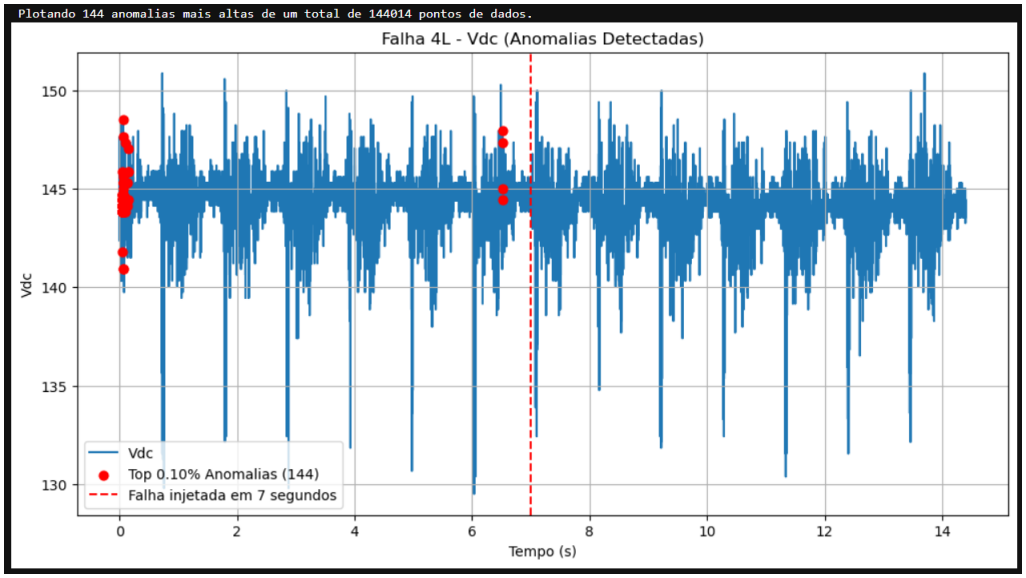
Como os autores LIU; TING; ZHOU (2008) e FOORTHUIS (2021) constataram, o algoritmo teve muita dificuldade em detectar as anomalias corretamente nos conjuntos de dados, de falhas 4, 6 e 7 de ambas as potências, que possuem anomalias com densidade local e do tipo enclausuradas. Demonstrado nos gráficos, figuras 34 e 35 abaixo:

Figura 34 – Top 0,1% de Anomalias Falha 7M



Fonte: (do Autor, 2024).

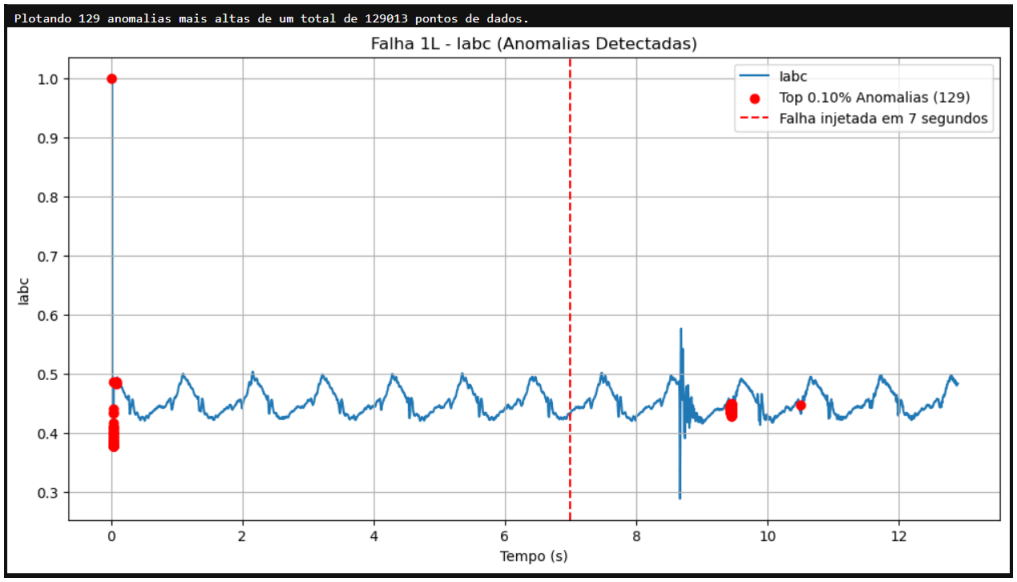
Figura 35 – Top 0,1% de Anomalias Falha 4L



Fonte: (do Autor, 2024).

Por último, o algoritmo teve dificuldade na falha 1 de potência limitada, que apesar de ser de densidade global e periférica, havia a suspeita das anomalias participarem de um "*cluster*"caso houvesse um comportamento semelhante em outras variáveis. Talvez por isso o algoritmo tenha tido dificuldade, como mostra a figura 36 abaixo:

Figura 36 – Top 0,1% de Anomalias Falha 1L



Fonte: (do Autor, 2024).

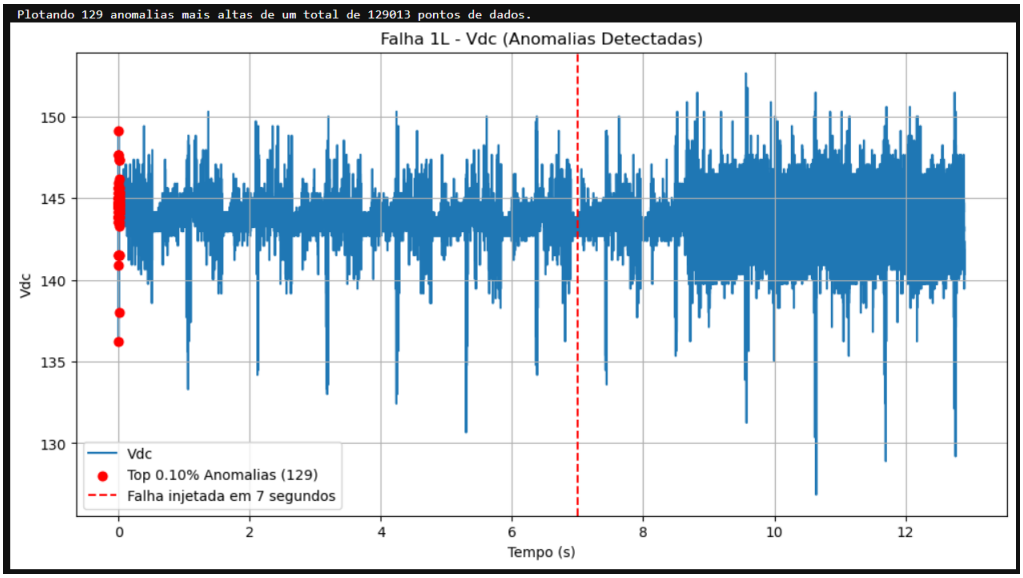
5.4.2 Floresta de Isolamento Extendida

Este algoritmo, diferente do padrão, possui um parâmetro chamado "*ExtensionLevel*"que traduz para "nível de extensão"que aumenta a complexidade das divisões

feitas pelo algoritmo, permitindo cortes em ângulos diferentes. Permite que as divisões sejam feitas por hiperplanos com inclinações aleatórias, não apenas paralelas ao eixo. Quanto mais próximo de 0, mais parecido com o funcionamento do algoritmo padrão, o limite sendo o número de dimensões do conjunto de dados, porém quanto maior o *extension level*, maior é o impacto no desempenho.

Com um *extension level* de 5, como é possível ver na figura 37 abaixo, o algoritmo determinou que o 0,1% de pontos com maior probabilidade de serem anomalias estão antes de 1 segundo e só a partir de 0,15% o mesmo considera anomalias após os 7 segundos na falha 1 limitada.

Figura 37 – Top 0,1% de Anomalias Falha 1L EIF



Fonte: (do Autor, 2024).

5.5 Tratamento dos Dados

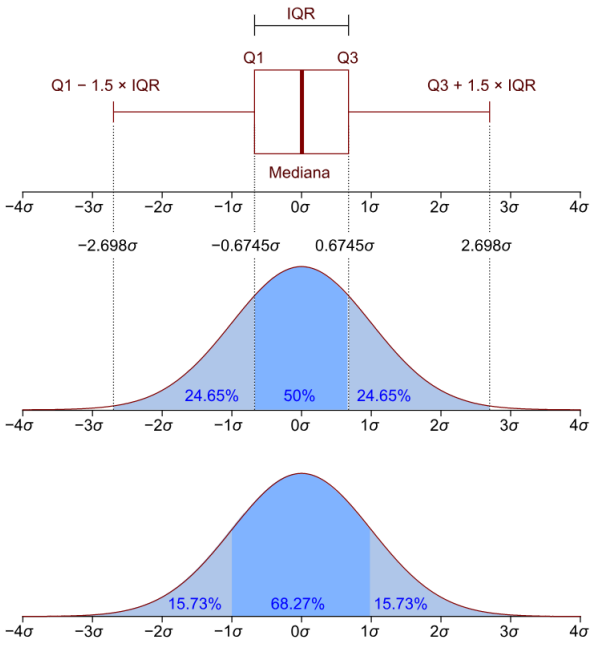
As variáveis "Iabc"e "Vabc"foram retiradas dos conjuntos antes do treinamento, pois de acordo com os autores: "As tensões da rede trifásica Uabc e as correntes Iabc apresentam padrões periódicos distorcidos e suas distribuições multimodais assimétricas não contribuem com informações de boa qualidade na fase de treinamento (BAKDI *et al.*, 2021). Também foram retirados dados duplicados e nulos para não interferirem no treinamento e na predição.

Como também explicado por (BAKDI *et al.*, 2021), os dados possuem uma grande quantidade de ruídos, neste caso é utilizado um filtro *Butterworth* de passa baixa (*Low-pass filter*) para filtrá-los. Ao utilizar um filtro de passa baixa é necessário escolher uma frequência de corte, a qual define o quanto dos dados será filtrado. Segundo (PROAKIS; MANOLAKIS, 1996) a frequência de corte é a frequência com a magnitude igual a -3dB. Para encontrar essa frequência de corte, é necessário aplicar a transformada de Fourier, convertendo os dados do domínio do tempo para o

domínio da frequência, o que pode ser feito utilizando a função ’np.fft.fft’ para calcular a transformada de fourier discreta, em seguida aplicar ’np.fft.fftfreq’ para obter as frequências correspondentes e por último analisar o espectro da frequência. Para o cálculo da transformada foi usado somente 0,2 porcento dos dados, pois a grande quantidade de pontos estava tornando o cálculo muito impreciso. Assim foi encontrada a frequência de corte de 34.84 Hz (figura) para o tipo "L"e 35.46 Hz para o tipo "M".

Por fim, por possuirem escalas muito diferentes entre si, os dados foram padronizados usando o algoritmo "*RobustScaler*", um escalonador de dados que padroniza os mesmos baseados na mediana e na amplitude interquartil, ilustradas na figura abaixo, que reduzem a influência de *outliers* com valores muito além do comum na padronização dos conjuntos de dados. Na figura, IQR significa interquartil.

Figura 38 – Função de densidade e diagrama de caixa de uma distribuição normal



Fonte: (CEPID NeuroMat).

6 APRESENTAÇÃO DOS RESULTADOS

Irá ser analisado como o algoritmo padrão e o extendido se saíram com os tipos diferentes de anomalia. Como este projeto não possui respostas, não sabemos quais pontos são com certeza anômalos fora analisando seu comportamento e distribuição incoerente contextualmente e do resto dos dados, para avaliar o resultado é necessário algumas suposições:

a) Todos os pontos classificados como anomalia antes de 7 segundos, ou seja, antes da injeção manual de falhas, são falsos positivos;

b) Todos os pontos classificados como anomalia após os 7 segundos, são verdadeiros positivos;

c) Todos os pontos não classificados como anomalias antes dos 7 segundos são verdadeiros negativos.

d) Não há como saber quais são os falsos negativos, então eles são considerados como 0.

Como não há como saber se são realmente anomalias, não há como medir os falsos negativos. Desta forma, sem uma avaliação prática ou contextual, sem injeção de respostas nos dados e sem criação de conjuntos novos, nos resta avaliar pela acurácia e precisão.

Avaliando com a precisão, que tem como objetivo minimizar falsos positivos, é uma métrica importante quando um alarme falso é muito custoso ou disruptivo. Dentro do contexto do trabalho, um sombreamento repentino em uma matriz fotovoltaica pode gerar um alarme falso, alertar técnicos erroneamente e o sistema pode ser interrompido para um diagnóstico desnecessário. A figura 2 abaixo ilustra como é calculada:

$$Preciso = \frac{VerdadeirosPositivos(TP)}{VerdadeirosPositivos(TP) + FalsosPositivos(FP)} \tag{2}$$

Já a acurácia, caracterizada pela equação da figura 3 abaixo, no caso deste projeto e boa parte de trabalhos que estudam de anomalias e *outliers* é desconsiderada, pois como um *outlier* é uma fração muito pequena do total dos dados, neste trabalho 0,1% e acurácia é basicamente a porcentagem de acerto, então mesmo um modelo simples que considerasse todos os pontos após 7 segundos como anomalias, teria 99,9% de acurácia. Sendo assim, para todos os casos apresentados, os resultados da acurácia são maiores ou iguais à 99,9%, então para fim de simplicidade nas matrizes de confusão apresentadas abaixo, a acurácia foi arredondada para 100%.

$$Acurcia = \frac{VerdadeirosPositivos(TP) + VerdadeirosNegativos(TN)}{TotaldeCasos} \tag{3}$$

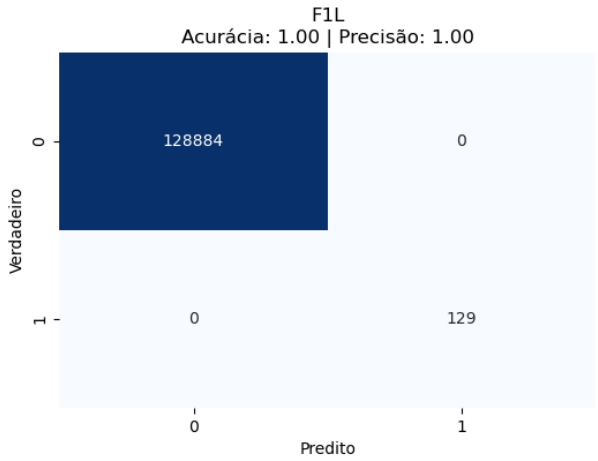
Como não sabemos quais são os falsos negativos e estamos os considerando como 0, a sensibilidade, também chamada de *recall*, terá o mesmo resultado que a precisão, como mostra a figura 4 abaixo:

$$Sensibilidade = \frac{VerdadeirosPositivos(TP)}{VerdadeirosPositivos(TP) + FalsosNegativos(FN)} \quad (4)$$

6.1 Floresta de Isolamento

Começando pelos resultados da Floresta de Isolamento padrão, que anteriormente obteve resultados satisfatórios com as anomalias de densidade global, falhas 1, 2 e 5 de potência máxima e falhas 2 e 3 de potência limitada. Na falha 1 limitada, onde anteriormente obteve resultado decepcionante, após o tratamente de dados seu resultado foi satisfatório. Na matriz de confusão 39 abaixo, vemos o resultado da previsão do algoritmo no conjunto de dados da falha 1 de potência limitada:

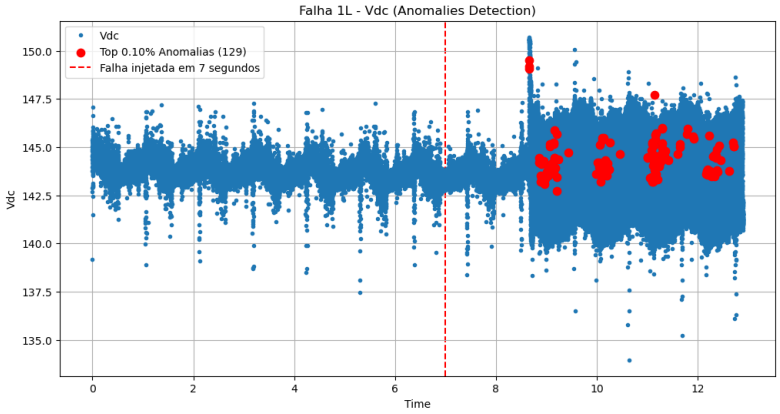
Figura 39 – Matriz de Confusão Floresta de Isolamento F1L



Fonte: (do Autor, 2024).

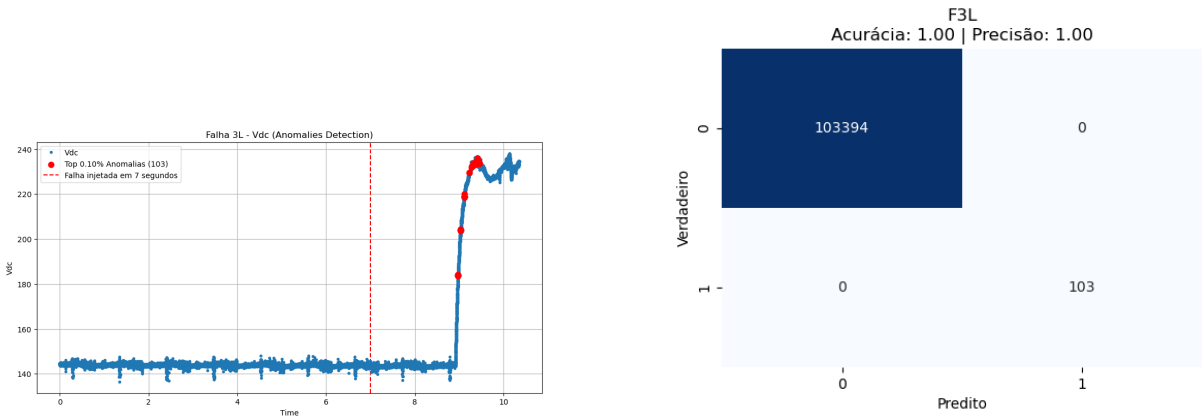
Abaixo, figura 40, as anomalias detectadas pelo mesmo no conjunto de mesma falha:

Figura 40 – Anomalias Detectadas F1L



Fonte: (do Autor, 2024).

Na falha 3 de potência máxima, apesar da falha acontecer e de haver um claro comportamento anormal antes dos 7 segundos como demonstra a figura abaixo, os autores BAKDI *et al.* (2021) não especificam em qual momento a falha foi injetada, o que compromete a avaliação final do algoritmo e por possuir uma classificação de anomalia comum e presente nos demais conjuntos de dados, o resultado final não será levado em conta no calculo das médias adiante. Abaixo 41 o gráfico da das detecções do algoritmo e a matriz do desempenho:

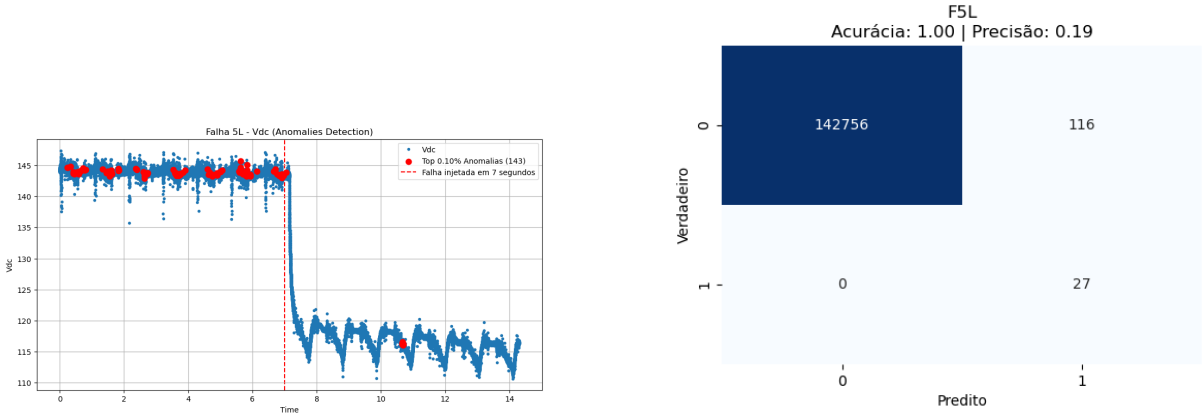


(a) Gráfico **(b) Matriz**

Figura 41 – Falha 3 limitada, desempenho e gráfico

Fonte: (do Autor, 2024).

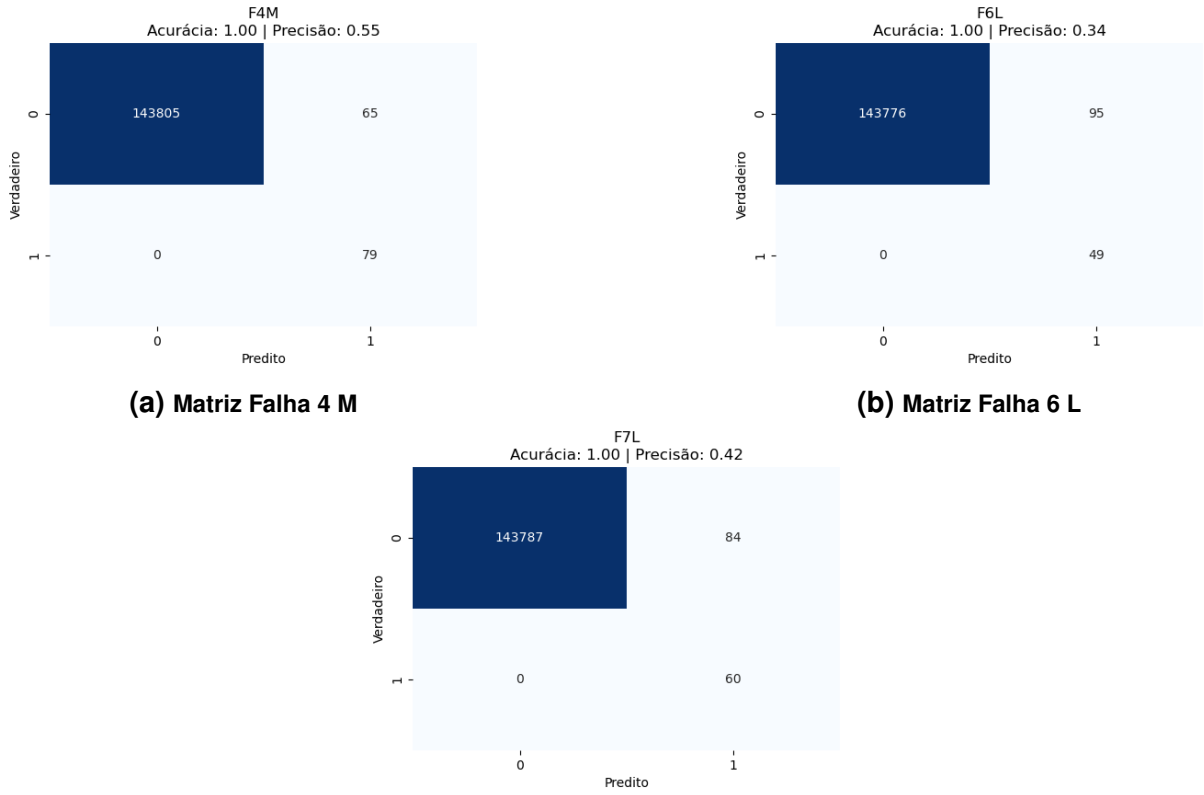
Surpreendentemente, o algoritmo obteve um desempenho decepcionante ao detectar as anomalais da quinta falha de potência limitada, como demonstram a matriz de confusão (42b) e o gráfico (42a), figura 42, abaixo:



(a) Gráfico **(b) Matriz**

Figura 42 – Falha 5 Limitada, Gráfico e Desempenho
Fonte: (do Autor, 2024).

Nas demais falhas, 4, 6 e 7, de densidade local e encalusuradas, o algoritmo se saiu como esperado, de modo não satisfatório. Abaixo 43 as matrizes de confusão demonstrado o desempenho do algoritmo am cada uma das falhas:



(a) Matriz Falha 4 M

(b) Matriz Falha 6 L

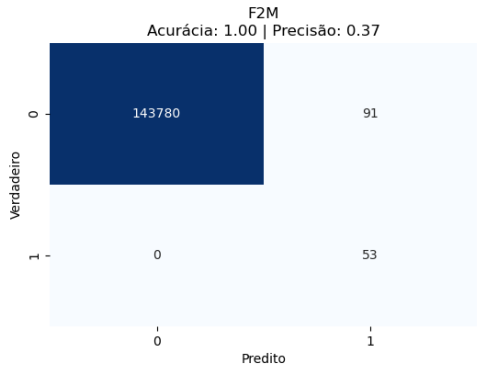
(c) Matriz Falha 7 M

Figura 43 – Matrizes de Confusão Falhas 4, 6 e 7
Fonte: (do Autor, 2024).

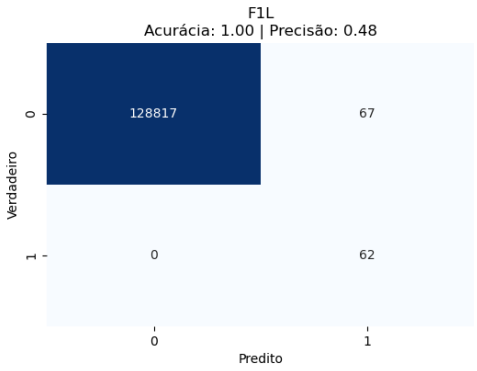
6.2 Floresta de Isolamento Extendida

Com a intenção de visualizar a influência das extensões no desempenho do algoritmo, foi testado com os níveis de extensão "*extension_level*"de 2, 5 e 10. O nível máximo de extensão N que o algoritmo aceita é igual ao número de variáveis N do conjunto de dados de treinamento $N = N$, porém no caso deste trabalho a variável "*Time*"foi utilizada como *index*, então $N = N - 1$.

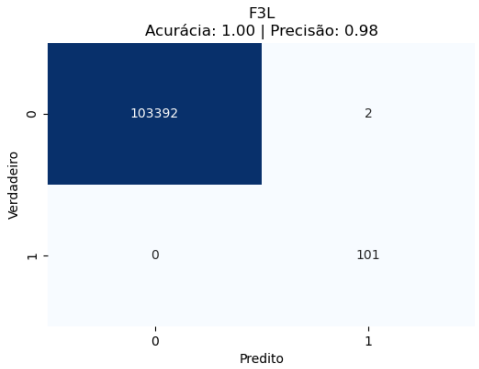
Nas falhas de densidade global, o algoritmo extendido se saiu no geral com o desempenho muito abaixo do algoritmo padrão, independente do seu nível de extensão, com algumas exceções onde ele alcança a precisão do mesmo. Nestas raras ocasiões em que isto acontece o número de extensões não encontra um padrão, o que sinaliza a possibilidade de estar relacionado a aleatoriedade do ângulo criado pela extensão e não ao desempenho do algoritmo extendido. Esta baixa repetibilidade pode indicar uma alta variância. As matrizes 44 abaixo demonstram o desempenho em falhas diferentes e extensões diferentes:



(a) Matriz Falha 2 M Extensão 2



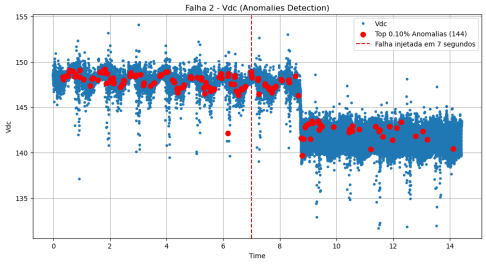
(b) Matriz Falha 1 L Extensão 5



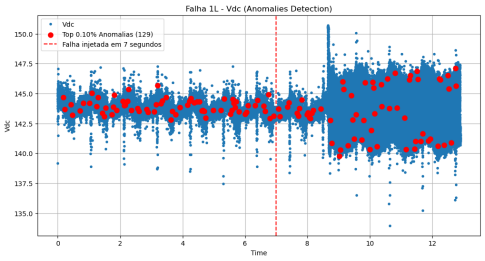
(c) Matriz Falha 3 L Extensão 10

Figura 44 – Matrizes de Confusão Falhas 2, 1 e 3, com Extensão e tipo Variados
Fonte: (do Autor, 2024).

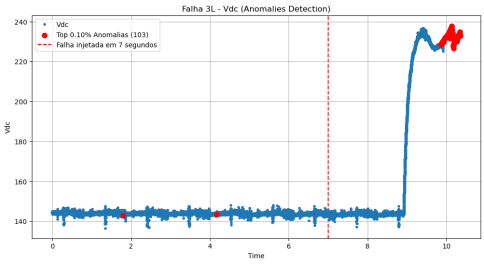
Os gráficos 45 abaixo demonstram as anomalias detectadas pelo algoritmo:



(a) Gráfico Falha 2 M Extensão 2



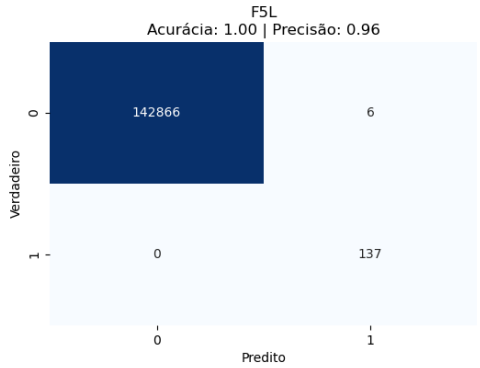
(b) Gráfico Falha 1 L Extensão 5



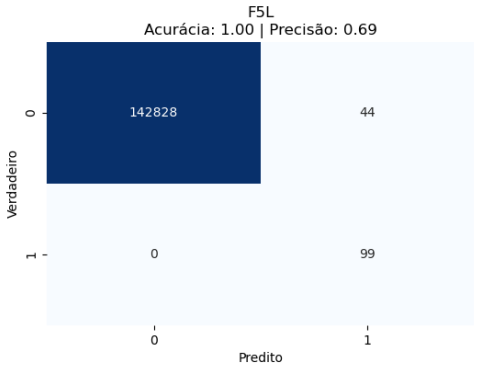
(c) Gráfico Falha 3 L Extensão 10

Figura 45 – Detecções das Anomalias nas Falhas 2, 1 e 3, com Extensão e tipo Variados
Fonte: (do Autor, 2024).

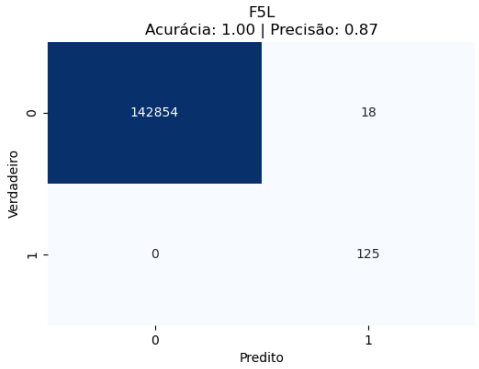
Na falha 5 do tipo limitada onde o algoritmo padrão obteve um desempenho ruim, as versões extendidas se saíram melhor, especialmente a com nível 2 de extensão. A precisão decresce de acordo com o crescimento do nível de extensão, o que pode representar um aumento na variância. As matrizes da figura 46 abaixo demonstram o desempenho de cada. As detecções das extensões 5 e 10 se distribuem de maneira quase identica as respectivas nos gráficos 45b e 45c.



(a) Matriz Falha 5 L Extensão 2



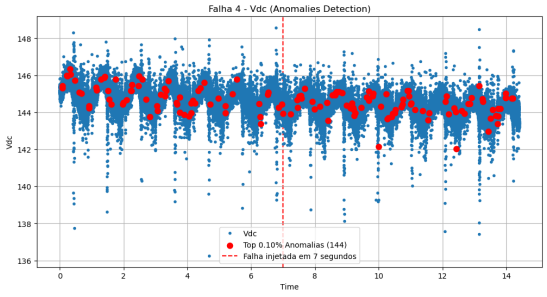
(b) Matriz Falha 5 L Extensão 5



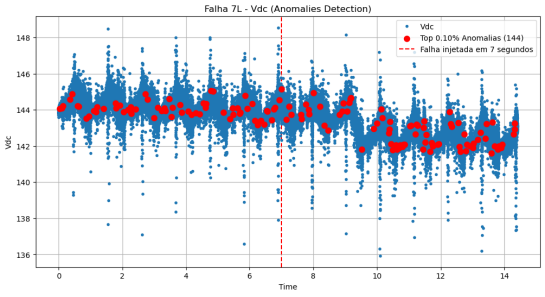
(c) Matriz Falha 5 L Extensão 10

Figura 46 – Matrizes de Confusão da Falha Limitada, com Extensões 2, 5 e 10
Fonte: (do Autor, 2024).

Nas demais falhas, 4, 6 e 7 de ambos os tipos, com anomalias de densidade local e enclausuradas, as versão extendidas obtiveram um desempenho melhor que a versão padrão, porém nada muito significativo, se mantendo em torno de 50% de precisão. Nos gráficos da figura 47 abaixo é possível determinar uma classificação quase que indiscriminada de pontos como anômalos, o que pode indicar um aumento de variância com a inclusão de extensões.



(a) Gráfico Falha 4 M

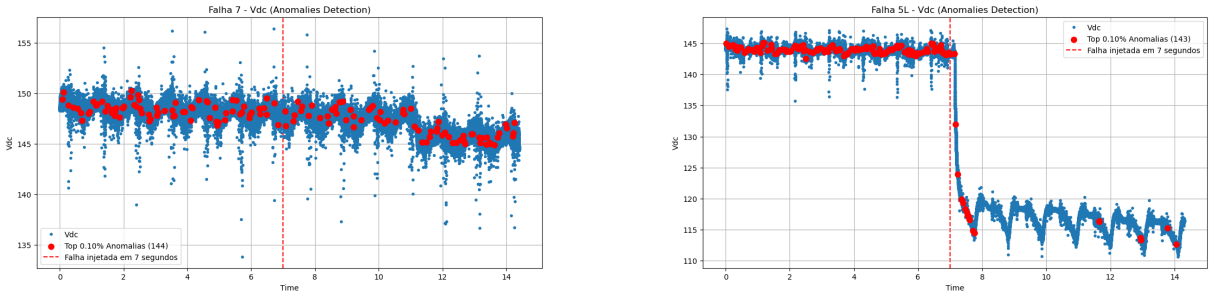


(b) Gráfico Falha 7 L

Figura 47 – Detecção em Anomalias de Densidade Local
Fonte: (do Autor, 2024).

Foi feita uma tentativa de diminuir a variância nos modelos extendidos diminuindo o "*sample_size*", equivalente ao "*max_sample*"da versão não extendida,

define a quantidade de amostras sendo retirada por vez para treinar cada estimador base, ou seja, quanto menor o valor deste parâmetro, mais o modelo se adequa ao conjunto de treinamento (aumenta o viés). Não foi observada uma melhora definitiva no desempenho do algoritmo, apenas uma aproximação dos resultados obtidos pela versão padrão nas falhas com anomalias de densidade global. A figura 48 abaixo demonstra esse desempenho.



(a) Gráfico Falha 7 M com Extensão 10 e *Sample* 128

(b) Gráfico Falha 5 L com Extensão 2 e *Sample* 128

Figura 48 – Detecção em Anomalias Extendida e *Sample_Size* Reduzido
Fonte: (do Autor, 2024).

6.3 Resultados Finais

Na figura (49) abaixo o resultado do modelo treinado com o algoritmo padrão de Floresta de Isolamento, vemos que o modelo conseguiu resultados bons nas falhas com anomalias de densidade global, com exceção da falha número 5. Nas demais falhas, talvez pelo viés intrínseco do algoritmo citado por HARIRI; KIND; BRUNNER (2021), o modelo foi incapaz de detectar as anomalias e acabou classificando até mesmo pontos normais como anômalos, possívelmente reproduzindo o que os autores LIU; TING; ZHOU (2008) chamam de "*masking*"e "*swamping*".

Figura 49 – Resultado Final Floresta de Isolamento

Floresta de Isolamento Padrão		
Precisão		
Falha	Limitada	Máxima
1	100,0%	100,0%
2	100,0%	100,0%
3	100,0%	0,0%
4	51,0%	55,0%
5	19,0%	100,0%
6	34,0%	50,0%
7	42,0%	44,0%
Média	51,0%	77,5%

Fonte: (do Autor, 2024).

Nas figura (50) abaixo vemos o desempenho dos modelos treinados com a versão extendida do algoritmo Floresta de Isolamento Extendida, onde os autores HARIRI; KIND; BRUNNER (2021) corrigem o viés citado anteriormente. Porém, observando os resultados percebe-se uma queda no desempenho geral dos modelos para com os conjuntos de falhas com anomalias de densidade global e uma estagnação no desempenho para com os conjuntos com anomalias de densidade local.

Floresta de Isolamento Extensão 2		
Precisão		
Falha	Limitada	Máxima
1	40,0%	100,0%
2	62,0%	37,0%
3	46,0%	0,0%
4	55,0%	55,0%
5	96,0%	86,0%
6	51,0%	58,0%
7	53,0%	49,0%
Média	53,0%	56,5%

(a) Nível de Extensão 2

Floresta de Isolamento Extensão 5		
Precisão		
Falha	Limitada	Máxima
1	48,0%	34,0%
2	62,0%	37,0%
3	58,0%	0,0%
4	51,0%	67,0%
5	69,0%	97,0%
6	51,0%	42,0%
7	74,0%	59,0%
Média	58,0%	50,5%

(b) Nível de Extensão 5

Floresta de Isolamento Extensão 10		
Precisão		
Falha	Limitada	Máxima
1	50,0%	100,0%
2	71,0%	51,0%
3	98,0%	0,0%
4	51,0%	49,0%
5	87,0%	87,0%
6	53,0%	53,0%
7	55,0%	50,0%
Média	55,0%	52,0%

(c) Nível de Extensão 10

Figura 50 – Resultado Final Floresta de Isolamento Extendida
Fonte: (do Autor, 2024).

Por fim, o parâmetro "*sample_size*"é reduzido na tentativa de que os modelos extendidos se adequem mais aos dados de treino e consigam discernir pontos normais de pontos anômalos nos conjuntos de dados de falhas. A figura abaixo (51) demonstram o desempenho e estagnação nas médias gerais:

Floresta de Isolamento Extensão 2 Sample 128		
Precisão		
Falha	Limitada	Máxima
1	88,0%	100,0%
2	97,0%	43,0%
3	29,0%	0,0%
4	44,0%	46,0%
5	15,0%	100,0%
6	51,0%	52,0%
7	55,0%	49,0%
Média	51,0%	50,5%

(a) Nível de Extensão 2 com *Sample* 128

Floresta de Isolamento Extensão 10 Sample 128		
Precisão		
Falha	Limitada	Máxima
1	52,0%	100,0%
2	44,0%	44,0%
3	31,0%	0,0%
4	52,0%	43,0%
5	17,0%	100,0%
6	50,0%	58,0%
7	58,0%	49,0%
Média	50,0%	53,5%

(b) Nível de Extensão 10 e *Sample* 128

Figura 51 – Floresta de Isolamento Extendida com *Sample_Size* Reduzido
Fonte: (do Autor, 2024).

7 CONSIDERAÇÕES FINAIS

Durante o trabalho, apresentou-se as etapas de desenvolvimento de um modelo detector de anomalias em instalações fotovoltaicas, utilizando como base o banco de dados criado pelos e a partir do experimento em laboratório dos autores (BAKDI *et al.*, 2021), desde um contexto introdutório até uma análise do desempenho dos modelos treinados e testados a partir dos dados utilizados. Em relação aos modelos desenvolvidos a partir do algoritmo *Isolation Forest*, os conjuntos de dados com anomalias de densidade global demonstraram a robustez e praticidade do algoritmo, lidando com um grande volume de dados de alta dimensão e obtendo resultados excelentes. Porém nos conjuntos com anomalias locais e enclausuradas, apesar de rápido e prático, é onde se torna clara a fraqueza do mesmo, incapaz de detectar estas anomalias e classificando dados normais indiscriminadamente, atingindo resultados em torno de 50% e por vezes até menores.A sua versão extendida *Extended Isolation Forest* foi usada para desenvolver modelos que pudessem superar as fraquezas da sua versão padrão, a versão extendida é onde as correções propostas pelos autores HARIRI; KIND; BRUNNER (2021) para as deficiências do algoritmo padrão descritas pelos mesmos tomam forma e buscam superar o viés do algoritmo original, mantendo sua robustez e praticidade. Os modelos desenvolvidos com a versão extendida se mostraram muito mais sensíveis as características de um grande conjunto de dados com alta dimensão, mal conseguindo obter resultados medianos nas anomalias de densidade local e raramente alcançando os resultados do algoritmo padrão nas anomalias de densidade global. Esta indiscriminação na classificação de *outiliers* por parte da versão extendida indica uma alta variância, ou seja, uma grade dificuldade do modelo se adequar aos conjuntos de treino, mesmo a tentativa de reduzir o *sample_size* pela metade para aumentar o viés, não obteve uma diferença significativa nos resultados.

A figura (52) abaixo demonstra as médias do desempenho de cada modelo, para as anomalias de densidade global e local, como também a média total do desempenho de cada um:

Figura 52 – Resultado Final

Média Final			
Modelo	**Global**	**Local**	**Total**
Padrão	88,4%	46,0%	67,2%
Extensão 2	66,7%	55,0%	60,9%
Extensão 5	57,9%	57,3%	57,6%
Extensão 10	77,7%	51,8%	64,8%
Ex2Samp128	67,4%	49,5%	58,5%
Ex10Samp128	55,4%	51,7%	53,5%

Fonte: (do Autor, 2024).

Com estes resultados, é possível concluir que para um contexto onde se há as necessidade de um algoritmo robusto, rápido e eficaz para lidar com anomalias de densidade global, *Isolation Forest* se destaca, porém deixa a desejar em relação a detecção de *outliers* de densidade local, principalmente quando lidando com conjuntos de dados de alta dimensão. Ainda assim é menos sensível aos mesmo em relação a sua versão extendida *Extended Isolation Forest*, a qual para alcançar resultados aceitáveis, precisa de dados mais tratados e adequados.

Por fim, o resultado final foi satisfatório, tendo estudado as características de diferentes anomalias e o comportamento de algoritmos tão distintos do comum, em aprendizado não-supervisionado, quando expostos as mesmas. Foi-se observado tanto as fraquezas descritas pelo autores, quanto seus valores ao lidar com conjuntos de dados desafiadores e custosos para muitos dos algoritmos da área.

O fato das tecnologias serem abertas e de simples utilização, como por exemplo as bibliotecas pandas e matplotlib, possibilita a realização deste tipo de estudo e experimento, podendo ser aplicado em outros tipo de instalações elétricas ou até mesmo em outro casos onde há anomalias e conjuntos de dados em séries temporais com alta dimensionalidade e grande número de instâncias, sendo estas os principais desafios do desenvolvimento deste tipo de estudo.

7.1 Sugestões para trabalhos futuros

Este trabalho exemplifica bem um grande desafio, não só para os algoritmos de aprendizado não supervisionado, mas todos os algoritmos de *machine learning* em geral, que é a "maldição da dimensionalidade". Trabalhos futuros podem explorar maneiras diferentes de abordar o problema, focando em algoritmos de alto desempenho em detecção de anomalias de densidade local ao invés de global e procurando maneiras de adequar os dados aos mesmo, através de praticas como redução de dimensionalidade, engenharia de variáveis, análise de importância das variáveis e sua correlações, etc.

Outra maneira de abordar este tipo de estudo é a adequação dos conjuntos de dados para os aprendizados semi-supervisionados e supervisionados através da injeção de rótulos das anomalias, que pode ser feito através de conhecimento contextual, reprodução prática ou pelo uso *auto-encoders*. Seja qual for a abordagem, há grande liberdade e possibilidade de aprendizado e desenvolvimento.

REFERÊNCIAS

BAKDI, A. *et al.* Real-time fault detection in pv systems under mppt using pmu and high-frequency multi-sensor data through online pca-kde-based multivariate kl divergence. *International Journal of Electrical Power and Energy Systems*, v. 125, p. 106457, 2021. ISSN 0142-0615. Disponível em: https://www.sciencedirect.com/science/article/pii/S0142061520300600. 12, 17, 31, 32, 33, 35, 47, 51, 59

BELLMAN, R. *Dynamic Programming*. [S.l.]: Dover Publications, 1957. ISBN 9780486428093. 26

BISHOP, C. *Pattern Recognition and Machine Learning*. Springer, 2006. (Information Science and Statistics). ISBN 9780387310732. Disponível em: https://books.google.com.br/books?id=qWPwnQEACAAJ. 18, 26

BOUMAN, R.; BUKHSH, Z.; HESKES, T. *Unsupervised anomaly detection algorithms on real-world data: how many do we need?* 2023. Disponível em: https://arxiv.org/abs/2305.00735. 27, 35

BURKOV, A. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. ISBN 9781999579517. Disponível em: https://books.google.com.br/books?id=0jbxwQEACAAJ. 18

FOORTHUIS, R. On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics*, Springer Science and Business Media LLC, v. 12, n. 4, p. 297–331, ago. 2021. ISSN 2364-4168. Disponível em: http://dx.doi.org/10.1007/s41060-021-00265-1. 24, 27, 45

FUCHS, E.; MASOUM, M. *Power Quality in Power Systems and Electrical Machines*. Elsevier Science, 2015. ISBN 9780128007822. Disponível em: https://books.google.com.br/books?id=ayC8rQEACAAJ. 17

GÉRON, A. *Mãos à obra: aprendizado de máquina com Scikit-Learn & TensorFlow*. Alta Books, 2019. ISBN 9788550803814. Disponível em: https://books.google.com.br/books?id=7Y7izQEACAAJ. 18, 19, 20, 21, 22, 23

GOLDSTEIN, M.; UCHIDA, S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLOS ONE*, Public Library of Science, v. 11, p. 1–31, 2016. Disponível em: https://doi.org/10.1371/journal.pone.0152173. 26

HARIRI, S.; KIND, M. C.; BRUNNER, R. J. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 33, n. 4, p. 1479–1489, abr. 2021. ISSN 2326-3865. Disponível em: http://dx.doi.org/10.1109/TKDE.2019.2947676. 41, 42, 56, 57, 59

IEA. *Why AI and energy are the new power couple, IEA, Paris*. 2023. https://www.iea.org/commentaries/why-ai-and-energy-are-the-new-power-couple. Accessed: 2023-11-30. 10

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: IEEE. *2008 Eighth IEEE International Conference on Data Mining*. [S.l.], 2008. p. 413–422. 24, 25, 45, 56

MCKINNEY, W. pandas: a foundational python library for data analysis and statistics. *Python High Performance Science Computer*, 01 2011. 43

MESSENGER, R.; VENTRE, J. *Photovoltaic Systems Engineering, Second Edition*. Taylor & Francis, 2004. ISBN 9780849317934. Disponível em: https://books.google.com.br/books?id=XiOeYrhyVIEC. 12, 13, 14, 15

PROAKIS, J.; MANOLAKIS, D. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice Hall, 1996. (Prentice-Hall International editions). ISBN 9780133942897. Disponível em: https://books.google.com.br/books?id=8471MAAACAAJ. 47

SMETS, A. *et al. Solar Energy: The Physics and Engineering of Photovoltaic Conversion, Technologies and Systems*. UIT Cambridge, 2016. ISBN 9781906860325. Disponível em: https://books.google.com.br/books?id=vTkdjgEACAAJ. 13, 15, 16, 17

STREETMAN, B. G. *Solid state electronic devices*. USA: Prentice-Hall, Inc., 1990. ISBN 0138229414. 12

WALT, S. van der; COLBERT, S. C.; VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 2, p. 22–30, mar. 2011. ISSN 1521-9615. Disponível em: http://dx.doi.org/10.1109/MCSE.2011.37. 43

ZIMEK, A.; SCHUBERT, E.; KRIEGEL, H.-P. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, v. 5, n. 5, p. 363–387, 2012. 26