

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
MINAS GERAIS - *CAMPUS* FORMIGA  
ENGENHARIA ELÉTRICA

Victor Gonçalves Lopes Cunha Amorim

**DESENVOLVIMENTO PRELIMINAR DE UM PACOTE PARA  
PREVISÃO AUTOMÁTICA DE POTÊNCIA SOLAR BASEADO EM  
DADOS HISTÓRICOS DO INMET**

Formiga - MG

2023

VICTOR GONÇALVES LOPES CUNHA AMORIM

**DESENVOLVIMENTO PRELIMINAR DE UM PACOTE PARA  
PREVISÃO AUTOMÁTICA DE POTÊNCIA SOLAR BASEADO EM  
DADOS HISTÓRICOS DO INMET**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus Formiga* para a obtenção do título de Engenheiro Eletricista.

**Orientador:** Prof. Me. José Antônio Moreira de Rezende

**Coorientador:** Prof. Dr. Reginaldo Gonçalves Leão Júnior

Formiga - MG  
2023

Amorim, Victor Gonçalves Lopes Cunha

A524d        Desenvolvimento preliminar de uma pacote computacional para  
previsão automática de potência solar baseado em dados históricos do INMET / Victor  
Gonçalves Lopes Cunha Amorim -- Formiga : IFMG, 2023.  
48p. : il.

Orientador: Prof. MSc. José Antônio Moreira de Rezende

Coorientador: Prof. Dr. Reginaldo Gonçalves Leão Júnior

Trabalho de Conclusão de Curso – Instituto Federal de Educação,  
Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1. Energia fotovoltaica. 2. Linguagem de programação JULIA. 3. Outliers.  
4. INMET. 5. Tratamento de dados. I. Rezende, José Antônio Moreira de. II. Leão  
Júnior, Reginaldo Gonçalves. III. Título.

CDD 621.3

VICTOR GONÇALVES LOPES CUNHA AMORIM

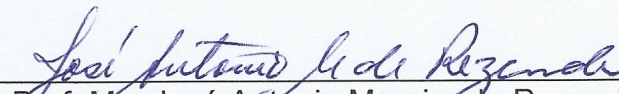
DESENVOLVIMENTO PRELIMINAR DE UM PACOTE PARA PREVISÃO  
AUTOMÁTICA DE POTÊNCIA SOLAR BASEADO EM DADOS HISTÓRICOS  
DO INMET

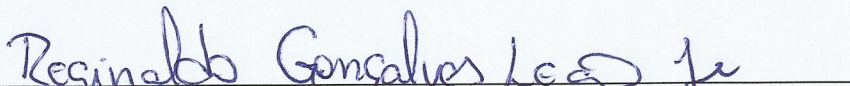
Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia  
Elétrica do Instituto Federal de Minas  
Gerais como requisito para obtenção do  
Título de Bacharel em Engenharia  
Elétrica.

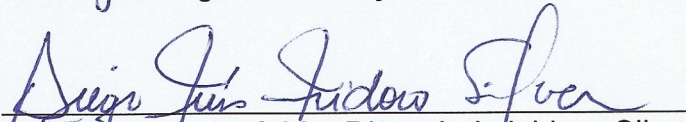
Avaliado em: 22 de junho de 2023.

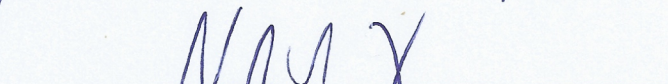
Nota: 81,6


BANCA EXAMINADORA

  
\_\_\_\_\_  
Prof. Me. José Antonio Moreira de Rezende (Orientador)

  
\_\_\_\_\_  
Prof. Dr. Reginaldo Gonçalves Leão Junior (Coorientador)

  
\_\_\_\_\_  
Prof. Me. Diego Luis Izidoro Silva

  
\_\_\_\_\_  
Prof. Dr. Elio da Costa Junior

  
\_\_\_\_\_  
Prof. Dr. Ricardo Carrasco Carpio

Este trabalho dedico à minha família, que sempre foram para mim sustento.

## **AGRADECIMENTOS**

Agradeço primordialmente à Deus, aquele que me capacita e me permite viver tudo isso, à minha família, pelo incansável apoio e sustento, e claro, ao meu orientador e co-orientador que voluntariamente se disponibilizaram a me ajudar a concluir este trabalho, assim como a inspiração como grandes profissionais. Ao grupo de pesquisa GESESC (Grupo de Estudos em Sistemas Energéticos e Simulação Computacional), por todo o suporte, auxílio, os quais foram essenciais para que fosse possível desenvolver este trabalho. Também à todos os colaboradores do IFMG Formiga que sustentam com trabalho e fazem crescer a instituição.

"Ama e faz o que quiseres. Se calares, calarás com amor; se gritares, gritarás com amor; se corrigires, corrigirás com amor; se perdoares, perdoarás com amor. Se tiveres o amor enraizado em ti, nenhuma coisa senão o amor serão os teus frutos."

Santo Agostinho

## RESUMO

Como alternativa de fonte de energia limpa, a instalação de usinas geradoras de energia elétrica que utilizam fontes renováveis tem conquistado papel de destaque na energética brasileira, já ocupando 48% da capacidade instalada. Dentre das várias existentes, a energia solar fotovoltaica vem crescendo exponencialmente, principalmente quando se diz sobre energia elétrica, através da implantação de painéis fotovoltaicos. No entanto, o incorreto dimensionamento, ou seja, projetado sem as devidas análises prévias, destes painéis ligados à rede pode causar perturbações e perdas técnicas na rede de distribuição já existente. Grande então é a necessidade de boas ferramentas preliminares, que facilitam a análise para o correto dimensionamento. Sendo assim, este trabalho apresenta o desenvolvimento de uma ferramenta preliminar para análise dos dados disponibilizados pelo INMET (Instituto Nacional de Meteorologia) que influenciam diretamente na geração de energia fotovoltaica e posteriormente o devido tratamento dos dados, principalmente a detecção de *outliers*, para que possam ser utilizados futuramente em um modelo de previsão de séries históricas. Todo o processo é desenvolvido para acontecer de forma automática através da linguagem de programação Julia. A metodologia foi desenvolvida em três partes: aquisição dos dados, tratamento dos dados, e detecção dos *outliers*. Com o método empregado foi possível visualizar graficamente os pontos detectados como *outliers*, assim como fazer uma filtragem para um conjunto de dados mais confiável sem a presença das anomalias.

**Palavras-chave:** Energia Fotovoltaica. Linguagem de programação Julia. *Outliers*.

## ABSTRACT

As a clean energy alternative, the installation of power plants that utilize renewable sources of electricity has gained prominence in Brazil's energy sector, already accounting for 48% of installed capacity. Among the various existing options, photovoltaic solar energy has been growing exponentially, especially in the context of electrical power, through the implementation of photovoltaic panels. However, incorrect sizing, meaning when designed without proper preliminary analysis, of these grid-connected panels can cause disturbances and technical losses in the existing distribution network. Therefore, there is a great need for reliable preliminary tools that facilitate analysis for proper sizing. Consequently, this study presents the development of a preliminary tool for analyzing data provided by INMET (National Institute of Meteorology) that directly influence the generation of photovoltaic energy, followed by appropriate data treatment, particularly the detection of outliers, which can be used in the future for historical series forecasting models. The entire process is designed to occur automatically using the Julia programming language. The methodology was developed in three parts: data acquisition, data treatment, and outlier detection. The employed method allowed for graphical visualization of the identified outlier points, as well as filtering the data set to obtain more reliable data without the presence of anomalies.

**Keywords:** Photovoltaic energy. Julia programming language. Outliers.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Gráfico da Capacidade Instalada de Geração Elétrica no Brasil em 2011 e 2021	13
Figura 2 – Implementação de sistema fotovoltaico <i>on-grid</i> .	26
Figura 3 – Fluxograma de desenvolvimento geral.	27
Figura 4 – Arquivos disponíveis para ano de 2022	29
Figura 5 – Dados disponíveis para ano de 2022 para cidade de Formiga	29
Figura 6 – Dados requeridos de Formiga no ano de 2010.	30
Figura 7 – Novo <i>dataset</i> redimensionado.	31
Figura 8 – Codificação para avaliação do comportamento de radiação solar acumulada anual.	32
Figura 9 – Gráfico para avaliação do comportamento de radiação solar acumulada anual	32
Figura 10 – Codificação para avaliação da correlação entre máxima temperatura diária e radiação solar acumulada	33
Figura 11 – Gráfico da correlação entre máxima temperatura diária e radiação solar acumulada	33
Figura 12 – Obtenção dos <i>datasets</i> de 2010 a 2023 para avaliação dos <i>outliers</i>	34
Figura 13 – Lista de nomes das colunas desconfigurada	35
Figura 14 – Lista de nomes das colunas configurada.	36
Figura 15 – Gráfico truncagem dados de pluviosidade diária acumulada	37
Figura 16 – Gráfico truncagem dados de radiação global diária acumulada	38
Figura 17 – Gráfico truncagem dados de amplitude térmica	38
Figura 18 – Gráfico dos dados de radiação solar global de 2010 até 2023	39
Figura 19 – Gráfico dos dados de pluviosidade de 2010 até 2023	40
Figura 20 – Gráfico dos dados de temperatura máxima diária de 2010 até 2023	41
Figura 21 – Gráfico dos dados de temperatura mínima diária de 2010 até 2023	42
Figura 22 – Dados concatenados de 2019 a 2023	43
Figura 23 – Gráfico de dados de radiação concatenados de 2019 a 2023	43
Figura 24 – Novo <i>dataset</i> com os <i>outliers</i> identificados	44
Figura 25 – Gráfico de dados de radiação com os <i>outliers</i> identificados	44
Figura 26 – Tabela de dados exportada com os dados filtrados	45

## LISTA DE ABREVIATURAS E SIGLAS

INMET	Instituto Nacional de Meteorologia
FRBNY	Federal Reserve Bank of New York
BNDES	Banco Nacional de Desenvolvimento Econômico e Social do Brasil
FAA	Administração Federal de Aviação dos Estados Unidos
PISA	<i>Programme for International Student Assessment</i>
KNN	<i>K-Nearest Neighbors Algorithm</i>
GPU	<i>Graphics Processing Unit</i>
KNNW	<i>K-Nearest Neighbors Weighted</i>
ODIN	<i>Outlier Detection using Indegree Number</i>
UTC	<i>Coordinated Universal Time</i>
GESESC	Grupo de Estudos em Sistemas Energéticos e Simulação Computacional
API	<i>Application Programming Interface</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Objetivos	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
2.1	Revisão da literatura	16
2.2	Linguagem Julia	17
2.2.1	<i>Breve histórico</i>	17
2.2.2	<i>Características basilares da linguagem Julia</i>	18
2.2.3	<i>Utilização da linguagem Julia</i>	19
2.2.4	<i>Comparações com outras linguagens</i>	20
2.3	Base de dados do INMET	20
2.4	Tratamento de dados	21
2.5	Detecção de <i>outliers</i>	22
2.6	Previsões de séries temporais	23
2.7	Sistemas fotovoltaicos	25
<b>3</b>	<b>METODOLOGIA</b>	<b>27</b>
3.1	Aquisição dos dados	28
3.2	Tratamento dos dados	31
3.3	Truncagem do <i>dataset</i>	33
3.4	Avaliação de séries e <i>outliers</i>	33
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>35</b>
4.1	Tratamento de problemas de codificação de caracteres	35
4.2	Dados truncados para futuro treinamento de redes neurais	37
4.3	Detecção dos <i>outliers</i>	37
<b>5</b>	<b>CONCLUSÃO</b>	<b>46</b>
5.1	Trabalhos futuros	46
	<b>REFERÊNCIAS</b>	<b>47</b>

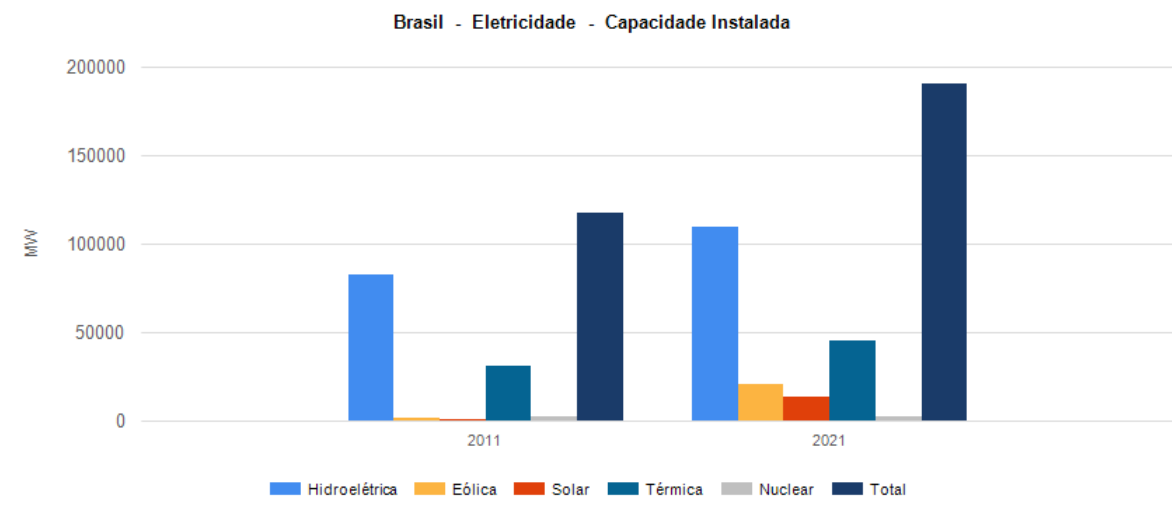
# 1 INTRODUÇÃO

As principais fontes de energia no mundo provém da queima de combustíveis fósseis, e estes contribuem significativamente na emissão de gases de efeito estufa como, por exemplo, o dióxido de carbono (CO<sub>2</sub>), metano (CH<sub>4</sub>), óxido nitroso (N<sub>2</sub>O), hexafluoreto de enxofre (SF<sub>6</sub>). Com o efeito, estes gases aprisionam parte do calor do sol e contribuem para o aquecimento da Terra. Dentro deste contexto, as fontes de energias sustentáveis entram como solução para diminuir tais efeitos, sendo que a produção de energia é essencial para a sociedade e seu desenvolvimento (TAVARES, 2023).

De acordo com o Departamento de Informações e Estudos Energéticos do Ministério de Minas e Energia (MME), a matriz energética renovável brasileira alcançou em 2022 a proporção de 48% em relação todas as fontes de energia presentes no país. Esta proporção significa que o indicador é três vezes a proporção mundial, mostrando a grande participação destas novas fontes na produção energética brasileira (ENERGIA, 2022).

Uma das fontes de energia renovável mais conhecidas é a energia solar. Esta pode ser utilizada tanto para produção de energia térmica (aquecimento de água, por exemplo), quanto para a produção de energia elétrica (energia fotovoltaica). Este recurso, como pode ser visto na Figura 1, cresceu significativamente em potência instalada de geração de energia elétrica por meio do Sol. Segundo SIE (2023), o Sistema de Informações Energéticas do Ministério de Minas e Energia, em 2011 a potência instalada de geração de energia solar era de 1 MW enquanto que em 2021 esse número subiu para 13.403,57 MW. Esse aumento se deve ao grande investimento nessa fonte de energia, devido à necessidade de uma fonte de energia que seja renovável, de fácil aplicação e investimento baixo.

Figura 1 – Gráfico da Capacidade Instalada de Geração Elétrica no Brasil em 2011 e 2021



Fonte: SIE (2023).

A inserção de painéis fotovoltaicos nas redes de distribuição de energia elétrica de baixa

tensão, no entanto, tem grande capacidade de impactar negativamente na qualidade da energia elétrica. Dependendo do seu correto ou não dimensionamento, o sistema fotovoltaico inserido na rede pode contribuir com perdas técnicas da rede de distribuição de energia elétrica (REINALDO; DUPCZAK; NETO, 2020) e (LEWANDOWSKI, 2023).

Segundo Neto (2016), a inserção de geração distribuída por meio de energia fotovoltaica na rede elétrica sem o correto dimensionamento pode causar alguns problemas, como por exemplo:

- **Fluxo de Potência em Sentido Reverso:** fluxo de potência gerado no sentido carga-fonte, contrariando o fluxo radial esperado pela rede de alimentação. Pode causar problemas de coordenação e operação com sistema de proteção das redes de distribuição.
- **Variações da tensão:** ainda que o sistema elétrico seja regulado, podem ser geradas tensões diferenciadas, para mais e para menos. Principalmente o aumento desta tensão pode causar danos aos componentes da rede, como por exemplo, banco de capacitores, autotransformadores.
- **Perdas elétricas:** com o aumento da carga no sistema, conseqüentemente aumenta-se o fluxo de potência, e como as perdas aumentam de acordo com o quadrado da corrente da carga, essa elevação acarretará no aumento quadrático das perdas.

Com isso, se faz necessário o desenvolvimento de modelos de previsão que possuem o objetivo de quantificar e analisar o comportamento do fornecimento desse tipo de energia renovável, a fim de garantir a confiabilidade e segurança da rede elétrica. Ferramentas especializadas em previsão de dados auxiliam na otimização do processo de dimensionamento, evitando problemas como os listados acima. (BASTOS *et al.*, 2020) (JESUS *et al.*, 2020).

Tais sistemas de energia renovável geralmente carregam consigo a utilização de tecnologia e automação, os chamados *Smart Grids* que, inseridos à rede, tendem a melhorar a qualidade da energia elétrica, através das informações obtidas, principalmente dados das condições climáticas da região, que influenciam diretamente na geração de energia solar fotovoltaica, por exemplo. Estes dados em conjunto com a predição de dados futuros ajudam a encontrar uma gestão otimizada da energia gerada (JESUS *et al.*, 2020).

## 1.1 Objetivos

Diante deste cenário, este trabalho tem como objetivo desenvolver uma ferramenta de extração e tratamento detalhado de dados de radiação solar, temperatura e pluviosidade, disponibilizados pelo Instituto Nacional de Meteorologia (INMET), preparando os dados para que sejam instrumento para uma confiável predição. Os dados utilizados para validação serão os fornecidos pela estação meteorológica automática de Formiga, sendo que a ferramenta permite acesso a

todas as cidades disponibilizadas pelo INMET, e o *software* será desenvolvido na Linguagem Julia.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Revisão da literatura

Dentro da literatura, diversos métodos para desenvolvimento deste tipo de ferramenta foram testados e utilizados, tanto do tratamento inicial dos dados, quanto também da sua aplicação para previsão.

Bacher, Madsen e Nielsen (2009) descrevem uma abordagem para previsão *online* para produção de energia em sistemas fotovoltaicos. Os autores utilizaram dados observados em uma janela de 15 em 15 minutos de energia solar de 21 sistemas fotovoltaicos localizados em uma vila da Dinamarca. A previsão de energia solar foi calculada utilizando modelos adaptativos de séries temporais lineares (autorregressivo e autorregressivo com entrada exógena). Os resultados mostram a diminuição de 35% do erro quadrático médio alcançado pelo modelo autorregressivo com entrada exógena em relação ao método de referência proposto.

Uma comparação de modelos de análise de dados, como detecção de *outliers*, entre o pacote *RegTools* da linguagem Julia e as ferramentas correspondentes da linguagem R foi desenvolvida por Liang (2015). Concluiu-se que o pacote de linguagem Julia é válida, pois ambas as ferramentas apresentam os mesmos resultados.

Marynych, Nazarenko e Khomenko (2017) analisaram séries temporais com grande quantidade de observações incoerentes e fora dos padrões utilizando pacotes da linguagem R. As descobertas revelaram as armadilhas da negligência de *outliers*, incluindo estacionariedade e especificação incorreta do modelo, estimativas de parâmetros tendenciosos, deterioração das propriedades dos resíduos e precisão da previsão dos modelos.

Sunderland *et al.* (2019) propõem um processo de avaliação da qualidade dos dados que enfatiza o uso de detecção multivariada de *outliers* para identificar erros e mostra que as abordagens univariadas são insuficientes.

Campos (2021) verificou que, quando se trabalha com um grande volume de dados, problemas como *outliers*, atributos vazios e replicados aparecem frequentemente. Estes problemas causam baixa qualidade da informação e reduzem do poder estatístico da base de dados. Com isso, desenvolveu um trabalho que avalia o impacto destes problemas na análise dos dados, especialmente os dados ausentes. Para isso, foi codificada uma versão do algoritmo *Logistic Regression* para realização dos testes e foram utilizados como base de dados os dados da Flor de Íris, que é amplamente reconhecida no meio acadêmico-científico. Ao analisar os resultados, a assertividade do algoritmo proposto diminui com a omissão dos dados.

Utilizando a linguagem de programação Julia, Laia (2021) aplicou diferentes técnicas de filtragem, exploração e classificação de dados em um conjunto de dados provenientes do PISA (*Programme for International Student Assessment*). Chegou-se à conclusão que é de suma importância na transformação de um aglomerado em um conjunto mais enxuto, que facilita a

visualização das características mais importantes para a análise desejada, assim, obtendo uma otimização e confiabilidade do processo.

Ao entender os problemas causados pelos *outliers*, Tushar *et al.* (2021) apresentam uma forma de identificar estes dados incoerentes, fazendo uma análise do padrão dos dados através da linguagem *Python* e suas respectivas bibliotecas, fazendo uma classificação de dois tipos de dados: *outliers* e dados normais. O algoritmo desenvolvido classificou com sucesso 3,27% do conjunto de dados como *outliers* e os eliminou da base de dados.

Oliveira (2023a) utilizou a linguagem de programação *Python*, com as bibliotecas como *Pandas*, *Numpy*, entre outras, para fazer a análise exploratória dos dados dos clientes de uma seguradora a fim de previamente verificar a probabilidade de ocorrência de sinistros, utilizando diferentes modelos de aprendizagem de máquina.

Foi desenvolvido por Rai, Shrivastava e Jana (2023) um modelo de previsão utilizando duas bases de dados em tempo real de duas usinas solares: uma localizada no deserto australiano e a outra em Delhi, Índia. O modelo proposto utiliza uma rede neural convolucional e em um modelo híbrido baseado em LSTM (*Long Short Term Memory*) bidirecional. Os resultados confirmam a eficácia do modelo proposto.

Sedai *et al.* (2023) realizaram um estudo profundo de métodos de previsão de produção de energia solar. São utilizados dados próximos a uma usina solar em Lubbock, Texas, nos Estados Unidos. Fazem testes com diversos métodos, dentre eles o modelo estatístico ARIMA (modelo auto-regressivo integrado de médias móveis), SVR (regressão de vetores de suporte), LSTM e GRU (Gated Recurrent Unit). Foi encontrado um modelo híbrido utilizando tais métodos como o com resultado com maior precisão.

É possível ver na literatura a utilização de métodos computacionais que sejam eficientes para o desenvolvimento do trabalho e chegada a resultados confiáveis.

## 2.2 Linguagem Julia

### 2.2.1 Breve histórico

Ao detectarem a necessidade de uma linguagem de programação dinâmica de alto nível combinada ao alto desempenho e eficiência (BEZANSON *et al.*, 2012a), quatro cientistas do *Massachusetts Institute of Technology* (MIT), Jeff Bezanson, Stefan Karpinski, Alan Edelman e Viral B. Shah, desenvolveram em 2012 a linguagem Julia (MAGALHAES; AVALOS; SANTOS, 2017).

Grande parte da programação científica requer estes dois benefícios, porém muito ainda é utilizado um esquema híbrido, com uma linguagem dinâmica fazendo a parte de alto nível, como MATLAB® e Octave, deixando o processamento complexo para C ou Fortran (MAGALHAES; AVALOS; SANTOS, 2017). Porém, tal recurso traz desvantagens, como por exemplo dificultar a

otimização de todo o programa. Com isso, Julia foi projetada para executar as modernas técnicas de execução de linguagens dinâmicas de forma eficiente (BEZANSON *et al.*, 2012a).

### 2.2.2 Características basilares da linguagem Julia

De acordo com Fieker *et al.* (2017), Bezanson *et al.* (2014) e Bezanson *et al.* (2012b), as características basilares que proporcionam tais benefícios desta linguagem incluem:

- **Compilação JIT (*Just-in-Time*):** que otimiza o código em tempo de execução, permitindo que os programas Julia sejam executados em velocidades próximas às de linguagens compiladas.
- **Sintaxe amigável:** A sintaxe de Julia é semelhante à de outras linguagens de programação populares, como Python, o que a torna fácil de aprender e ler. Ela possui uma notação matemática natural e permite expressar algoritmos de forma clara e concisa.
- **Suporte a metaprogramação:** Julia possui recursos avançados de metaprogramação, permitindo a criação e manipulação de código em tempo de execução. Isso facilita a escrita de código genérico e flexível, adaptável a diferentes tipos de dados e operações.
- **Ecossistema extenso:** Julia possui um ecossistema rico de pacotes e bibliotecas para diversas áreas, como computação científica, aprendizado de máquina, estatística, visualização de dados, processamento de sinais, entre outros. O gerenciador de pacotes integrado, chamado de "Pkg", torna fácil a instalação e atualização de pacotes.
- **Tipagem dinâmica:** Julia é uma linguagem dinamicamente tipada, o que significa que as variáveis não precisam ter seus tipos especificados explicitamente. No entanto, também suporta a especificação opcional de tipos para melhorar o desempenho em código numérico.
- **Integração com outras linguagens:** Julia pode ser facilmente integrada com código escrito em outras linguagens, como C, Python e R. Isso permite aproveitar bibliotecas existentes nessas linguagens e tirar proveito de funcionalidades específicas de cada uma delas.
- **Despacho múltiplo dinâmico:** O despacho múltiplo é uma forma de polimorfismo que permite que funções tenham diferentes implementações com base nos tipos dos argumentos passados. Em Julia, o despacho é feito em tempo de compilação, permitindo que as funções sejam otimizadas para os tipos específicos dos argumentos em tempo de execução. Nesta linguagem, as funções podem ser definidas de forma genérica, ou seja, sem especificar os tipos de dados exatos dos argumentos. Quando este tipo de função é chamada, o mecanismo de despacho múltiplo define a implementação mais eficiente para os tipos de argumentos fornecidos. Ele seleciona automaticamente a versão mais apropriada da função, com base nos tipos, e a chama. Isso permite um polimorfismo eficiente e flexível.

Dentre tais características, os desenvolvedores da linguagem destacam como essencial para o entendimento do seu funcionamento a funcionalidade do despacho múltiplo dinâmico:

*O principal meio de abstração de Julia é o despacho dinâmico múltiplo. Grande parte de uma linguagem consiste em mecanismos para selecionar código a ser executado em diferentes situações — desde a seleção de método até a seleção de instrução. Utilizamos apenas o despacho dinâmico múltiplo para esse propósito, o qual é possível através de regras de despacho suficientemente expressivas. (BEZANSON et al., 2012a).*

Esta característica permite executar as instruções do código em ordem (*in-order execution*) ou fora de ordem (*out-of-order execution*). Ou seja, utilizam o *hardware* para execuções em paralelo no tempo de execução. Importante frisar que somente a execução permite essa dinamicidade, sendo que a entrega dos resultados é sempre em ordem. É presente também em outras linguagens, como *Common Lisp*, *Python*, com a adição de bibliotecas específicas (CALÇADA, 2020).

### 2.2.3 Utilização da linguagem Julia

Diante deste cenário, grandes empresas, centros de pesquisas e universidades têm aplicado a linguagem Julia, seja no desenvolvimento de seus produtos, quanto como na implementação e otimização de seus processos. De acordo com Magalhaes, Avalos e Santos (2017) e JuliaHub (2023), são exemplos:

- *Federal Reserve Bank of New York (FRBNY)*: no desenvolvimento de modelos da economia americana;
- *BlackRock, Inc*: empresa norte-americana, para análises de dados temporais e aplicações em *big data*, desenvolvendo o produto Alladin, que é um sistema eletrônico para gerenciadores de investimento;
- Banco Nacional de Desenvolvimento Econômico e Social do Brasil (BNDES): no modelo matemático para realização de cálculos que incluem transações de classificação, dividendos e acumulações;
- Aviva: empresa de seguros europeia, no sistema de modelagem de riscos para economia global;
- Administração Federal de Aviação dos Estados Unidos (FAA): no sistema de prevenção de colisão aérea.
- A *Zipline*: para simulação de aeronaves para fornecer suprimentos médicos de emergência que salvam vidas via drone na África e no mundo;

- *QuantEcon*: empresa que tem como fundador o vencedor do prêmio Nobel de economia em 2011, Thomas J. Sargent, no desenvolvimento de modelos macroeconômicos;
- *United Therapeutics Corporation*: na construção de um modelo computacional do pulmão para desenvolver tratamentos para doenças raras, incluindo medicamentos e a sua ação no organismo;
- PSR: empresa global de consultoria, análise e tecnologia de eletricidade e gás natural, na simulação, análise e planejamento do mercado de energia;
- Pesquisadores do Reino Unido: no modelo de simulação de crescimento de tumores;
- Pesquisadores da Universidade da Califórnia em Berkeley: no controle de carros de corrida autônomos.

#### 2.2.4 Comparações com outras linguagens

Um estudo realizado por Axillus (2020), do Instituto de Tecnologia Blekinge, na Suécia, busca fazer algumas comparações pontuais da linguagem Julia com a linguagem *Python*, esta considerada a mais popular quando se trata de aprendizado de máquina e prototipagem. Para as comparações, em termos de tempo de execução, o estudo realizou testes em processamento de imagem com redes neurais profundas aceleradas por GPU e classificação com k-vizinhos mais próximos (*K-Nearest Neighbors Algorithm - KNN*). Os resultados apontam Julia superando *Python* em aproximadamente 1,25 a 1,5 vezes para o primeiro teste, já para o segundo teste, Julia supera *Python* em cinco de oito medições realizadas.

Antonio (2019) faz uma comparação mais específica entre Julia, *Python* e R, para análise de valores de pressão arterial de pacientes, realizando uma pesquisa científica para obtenção desta comparação. Com os resultados encontrados constata-se que a linguagem Julia supera *Python* e R devido à sua flexibilidade, abordagens amigáveis à máquina, bem como aos recursos de execução rápida.

### 2.3 Base de dados do INMET

A base de dados do INMET (Instituto Nacional de Meteorologia) consiste em um conjunto de informações meteorológicas coletadas de suas estações distribuídas por todo o território brasileiro. Essa base de dados contém uma ampla variedade de informações, incluindo dados históricos, dados em tempo real e previsões meteorológicas (INMET, 2023).

Os dados históricos geralmente abrangem um longo período de tempo e incluem informações de temperatura, precipitação, umidade, vento, radiação solar, pressão atmosférica, entre outros parâmetros meteorológicos. Esses dados históricos são usados para análises climatológicas, estudos científicos e modelagem de previsão do tempo.

Os dados são coletados em intervalos regulares, geralmente a cada hora, e são armazenados em bancos de dados do INMET. Essas informações são então processadas e disponibilizadas ao público de diferentes formas, como por meio do *website* do INMET, serviços de dados meteorológicos em tempo real ou histórico, e também podem ser acessadas por meio de APIs (*Application Programming Interface*) para integração com outros sistemas (INMET, 2023).

Os dados das estações meteorológicas do INMET são utilizados para diversos fins, como previsão do tempo, monitoramento climático, estudos científicos, suporte à agricultura, gestão de recursos hídricos, entre outros. Eles são valiosos para entender as condições meteorológicas locais, regionais e nacionais, auxiliando na tomada de decisões em diversas áreas que dependem do clima e das condições atmosféricas.

## 2.4 Tratamento de dados

A Ciência de Dados, de acordo com Azevedo (2019) e Oliveira (2023b), abrange um conjunto de princípios, técnicas, algoritmos de processos para a extração de padrões não evidentes em grandes conjuntos de dados. Os autores sintetizam ainda dizendo que é a ciência que estuda informações, sua captura, transformação, geração e análise.

Segundo Vieira (2021), a qualidade dos dados é o principal fator que influencia na robustez e confiabilidade dos resultados de qualquer análise realizada. Quando se trabalha com um banco de dados com grande quantidade de informações e, principalmente, quando tais dados provêm de medições automáticas, possivelmente o banco de dados pode conter, por exemplo, alguns valores equivocados em função de erros de medida dos equipamentos e dados faltantes por motivo de interrupções de funcionamento, o que prejudicam uma análise a posteriori. Olson e Delen (2008) também descrevem importantes técnicas para o tratamento dos dados:

- **Limpeza dos dados:** consiste na identificação de linhas ou colunas com dados faltantes, tendo como solução a exclusão destas linhas ou colunas ou simplesmente o preenchimento das lacunas faltantes com valores estimativos adequados. Isto também pode acontecer para os *outliers*, que são valores destoantes dos demais que vêm a distorcer a análise. Estes valores podem ser excluídos ou substituídos.
- **Normalização:** transformar as escalas para uma escala comum, a fim de evitar escalas dominantes durante a análise e prejudicar a visualização e análise das demais.
- **Padronização:** deixar as variáveis com média nula e desvio padrão unitário, a fim de facilitar a comparação e análise das variáveis.
- **Agregação e resumo:** separar dados de acordo com características comuns e calcular estatísticas, como por exemplo média, mediana, mínimo, máximo, etc., a fim de obter uma visão ampla dos dados.

- **Discretização:** transformar dados contínuos em dados discretos, o que permite serem manipulados assim como os outros.
- **Tratamento de *labels* inconsistentes:** lidar com *labels*, ou seja, nomes inconsistentes, como mesclagem de categorias semelhantes ou padronização de nomes para evitar duplicações.
- **Redução de dimensionalidade:** literalmente selecionar e excluir os dados inconsistentes, sem sentido para o desenvolvimento.

A técnica para corrigir tais problemas listados acima, é chamada de tratamento de dados ou também de pré-processamento. Este processo é considerado semiautomático, sendo que requer do analista dos dados uma análise crítica para verificar quais dados precisam ser pré-processados, corrigidos, readequados, a neles aplicar a técnica viável para solucionar tais problemas.

Ainda de acordo com Vieira (2021), este processo consiste em três fases:

1. Compreensão do problema: entender de fato qual matéria prima precisa para resolver o problema estudado.
2. Compreensão dos dados: verificar de que forma os dados são obtidos "*in natura*" e como está estruturado.
3. Preparação dos dados: fazer as devidas correções necessárias e avaliar de fato quais dados serão úteis, para não carregar informações desnecessárias e atrapalhar o desempenho do processo.

Sendo assim, é imprescindível esta etapa para quaisquer análises a serem feitas posteriormente, como por exemplo, previsões de séries temporais.

## 2.5 Detecção de *outliers*

Em um estudo estatístico aplicado a um grande conjunto de dados, geralmente encontram-se dados que se desviam do padrão, ou seja, que seu comportamento se distancia consideravelmente dos demais dados. O surgimento de tais anomalias podem ter diversas causas (FREITAS *et al.*, 2019).

Existe, em determinados casos, a variação natural da distribuição dos dados, onde determinada característica ou atributo dos dados de fato ocorre por se diferenciar dos demais, como por exemplo o valor de um automóvel muito mais caro em relação aos demais. Também pode ocorrer o erro de medida e coleta dos dados, seja por um erro humano ou erro de um dispositivo automático com sensores com defeito ou desconfigurados, gerando assim dados inconsistentes (FREITAS *et al.*, 2019).

Campos (2015) descreve alguns métodos eficientes para detecção destes dados fora dos padrões:

- **KNN (*k-nearest neighbors*)**: este método considera a distância de um ponto  $p$  para seu  $k$ -ésimo vizinho mais próximo. Quanto maior esta distância, maior é o potencial deste dado ser um *outlier*. O valor de  $k$ , ou seja, a quantidade de vizinhos que serão analisados em relação ao ponto  $p$ , determinam a robustez do algoritmo. Quanto maior a quantidade analisada, mais confiável, quanto menor, mais sensível às discrepâncias se torna o algoritmo.
- **KNNW (*k-Nearest Neighbors Weighted*)**: carrega a mesma ideia do método KNN, porém, de forma ponderada. Ao invés de considerar a distância  $p$  até o  $k$ -ésimo vizinho mais próximo, este utiliza a distância de todos os pontos vizinhos até o  $k$ -ésimo vizinho. Este método evita que dois pontos com igual distância possuam o mesmo peso no final.
- **ODIN (*Outlier Detection using Indegree Number*)**: é um método de detecção de *outliers* baseado no número de grau de entrada em um grafo. Esse método utiliza um grafo direcionado, onde cada nó representa uma instância de dados e as arestas representam relações entre essas instâncias. Para cada nó no grafo, é calculado o número de grau de entrada, ou seja, o número de arestas que apontam para esse nó. Um limiar é definido para determinar quando uma instância é considerada um *outlier* com base no número de grau de entrada. Se o número de grau de entrada de uma instância for menor que o limiar estabelecido, a instância será classificada como um *outlier*.

## 2.6 Previsões de séries temporais

De acordo com Paula (2023), séries temporais são um conjunto de dados que são resultado de uma observação de uma determinada variável ordenadas no tempo, respeitando períodos regulares de medições. Os estudos estatísticos baseados no comportamento histórico destas séries temporais que, por sua vez, buscam prever o comportamento de tal série para algum tipo de análise, é a chamada previsão de séries temporais.

A previsão de séries temporais tem como objetivo, através de dados históricos, prever/estimar o comportamento destes mesmos dados da série temporal. Este procedimento pode ser feito através de modelos matemáticos, métodos estatísticos ou também técnicas de *machine learning* (aprendizagem de máquina). Importante verificar se os dados que serão trabalhados exibem uma dependência temporal, com dados passados que influenciam nos dados futuros. Estas dependências permitem a captação de padrões entre os dados (RAMOS, 2010).

Os métodos existentes abrangem desde técnicas mais simples, como modelos autorregressivos e média móvel por exemplo, até modelos mais robustos, como modelos *ARIMA* *Auto Regressive Integrated Moving Average* e redes neurais recorrentes (RNN). São alguns modelos de previsão de séries temporais (RIO, 2023):

- **Modelo Naive-Bayes:** é um modelo mais simples, sendo que toma como base para definição do valor futuro, o último valor conhecido. Como exemplo: um conjunto de dados de temperatura média de segunda a sexta em uma determinada cidade é de 28°C, 25°C, 26°C, 29°C, 30°C, este método entenderá que no sábado a temperatura será de 30°C, devido à última temperatura medida.
- **Modelo Autoregressivo (AR):** os modelos autorregressivos são utilizados quando é viável assumir que o valor futuro da série temporal seja dependente imediatamente do valor anterior, porém, acrescido de um erro aleatório. Esta relação entre o valor passado e futuro pode não ser suficientemente linear para utilizar este método, no caso, em séries temporais com padrões mais complexos.
- **Suavização exponencial (*Exponential Smoothing*):** modelo que utiliza pesos decrescentes dos valores atuais até aos valores mais antigos, ou seja, dando maior peso e importâncias aos valores mais atuais. Atualiza automaticamente a previsão a medida que novos dados são adicionados à série temporal estudada.
- **Média Móvel (MA):** apesar do nome, não se refere ao conceito literal de média móvel. Os valores da série, neste modelo, se referem a uma combinação linear das perturbações.
- **Modelo *AutoRegressive Moving Average* (ARMA):** que significa combinação de um modelo Autorregressivo(AR) e de média móvel(MA). Ou seja, possuem uma estrutura de dependência com valores passados, assim como também em relação às perturbações. São adequados para séries temporais estacionárias e são capazes de capturar dependências lineares entre os valores passados e futuros.
- **Modelo *AutoRegressive Integrated Moving Average* (ARIMA):** os modelos anteriormente descritos são utilizados apenas para descrever séries históricas estacionárias, sendo necessário retirar características como a tendência, por exemplo. Neste modelo, componente de diferenciação da série é introduzido para permitir que a série se torne estacionária, além de combinar os componentes autoregressivos (AR), que modelam a relação entre um valor e seus valores anteriores, com componentes de média móvel (MA), que modelam a relação entre o erro e seus valores anteriores.
- **Redes neurais recorrentes (RNN - *Recurrent Neural Networks*):** são redes neurais projetadas para trabalhar com sequências. Elas possuem uma capacidade intrínseca de lidar com dependências temporais e podem capturar relações complexas entre os dados passados e futuros. Variantes como *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU) são amplamente utilizadas nessa área.

Os métodos ARMA e ARIMA, para que suas aplicações sejam válidas, necessita que as séries sejam estacionárias. Isso acontece pelo fato dos modelos assumem que a relação entre os valores passados e futuros das séries temporais permanecem constantes ao longo do tempo.

Para o método ARIMA, especificamente, existem algumas razões pela qual tornar as séries estacionárias ser um passo tão importante:

- **Relações temporais constantes:** O modelo ARIMA assume que a relação entre os valores passados e futuros é constante ao longo do tempo. Isso significa que as propriedades estatísticas da série, como a média e a variância, não devem mudar com o tempo. Se a série não for estacionária, essas relações podem ser afetadas por tendências, sazonalidades ou outros padrões que violam a suposição do modelo.
- **Estimação dos parâmetros:** A estimação dos parâmetros do modelo ARIMA é mais estável e confiável em séries estacionárias. Em séries não estacionárias, os parâmetros podem ser estimados de forma imprecisa ou inconsistente, o que pode levar a previsões incorretas.
- **Previsões mais precisas:** A estacionariedade permite que o modelo ARIMA capture os padrões e relações dos valores passados de forma mais precisa. A remoção de tendências e sazonalidades não significativas ajuda a revelar os padrões subjacentes que são relevantes para a previsão futura. Isso resulta em previsões mais precisas e confiáveis.

## 2.7 Sistemas fotovoltaicos

Sistemas fotovoltaicos são sistemas que convertem a energia solar em eletricidade por meio de células solares fotovoltaicas. Essas células são compostas por materiais semicondutores, como o silício, que absorvem a luz solar e geram corrente elétrica quando expostos à radiação solar.

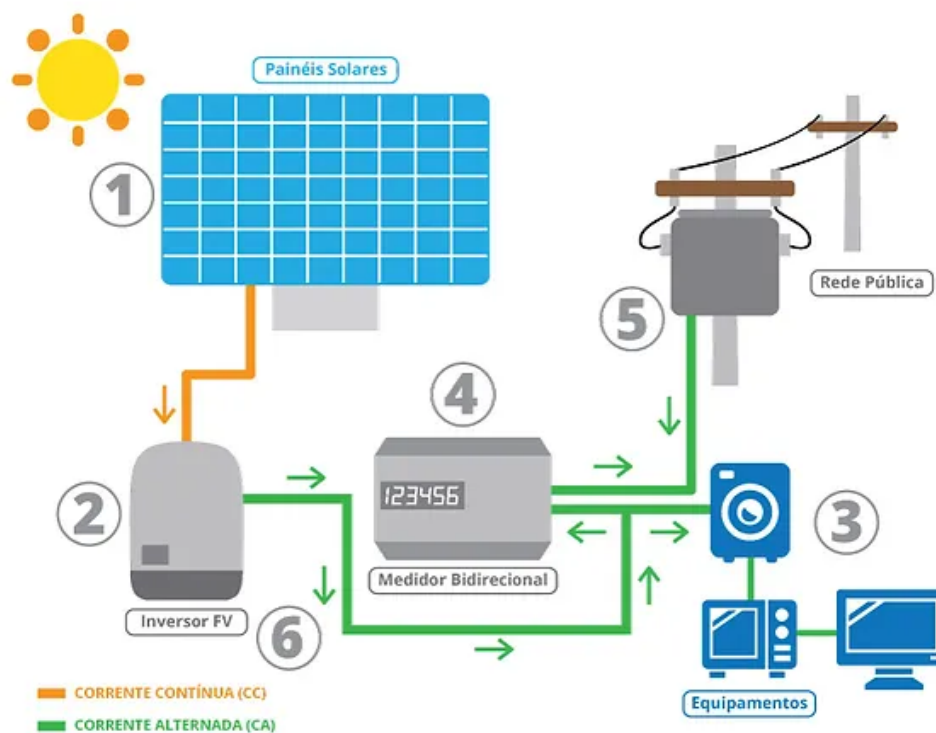
De acordo com Green *et al.* (2006), vários fatores podem influenciar o desempenho e a eficiência dos sistemas fotovoltaicos:

- **Irradiação solar:** A quantidade de radiação solar incidente na superfície dos painéis solares é um fator crucial. Regiões com maior incidência solar terão maior geração de energia fotovoltaica. A latitude, a altitude, a orientação e a inclinação dos painéis também afetam a quantidade de radiação solar recebida.
- **Temperatura:** A temperatura ambiente afeta o desempenho das células solares. O aumento da temperatura reduz a eficiência da conversão da luz solar em eletricidade. Portanto, é importante considerar a dissipação de calor e a ventilação adequada nos sistemas fotovoltaicos.
- **Sombreamento:** A presença de sombras sobre os painéis solares pode diminuir significativamente a geração de energia. Mesmo uma pequena área sombreada em um painel pode afetar negativamente o desempenho de todo o sistema. Portanto, a localização e o posicionamento dos painéis devem ser planejados para evitar sombreamento.

- **Pluviosidade:** Durante períodos chuvosos ou nublados, a cobertura de nuvens pode bloquear a luz solar direta, reduzindo a quantidade de radiação solar incidente nos painéis solares. Isso resulta em uma diminuição na geração de energia durante esses períodos.

A implementação de um sistema fotovoltaico *on-grid*, ou seja, o sistema que está ligado diretamente à rede elétrica, que é o caso de estudo deste trabalho, é mostrado na Figura 2.

Figura 2 – Implementação de sistema fotovoltaico *on-grid*.



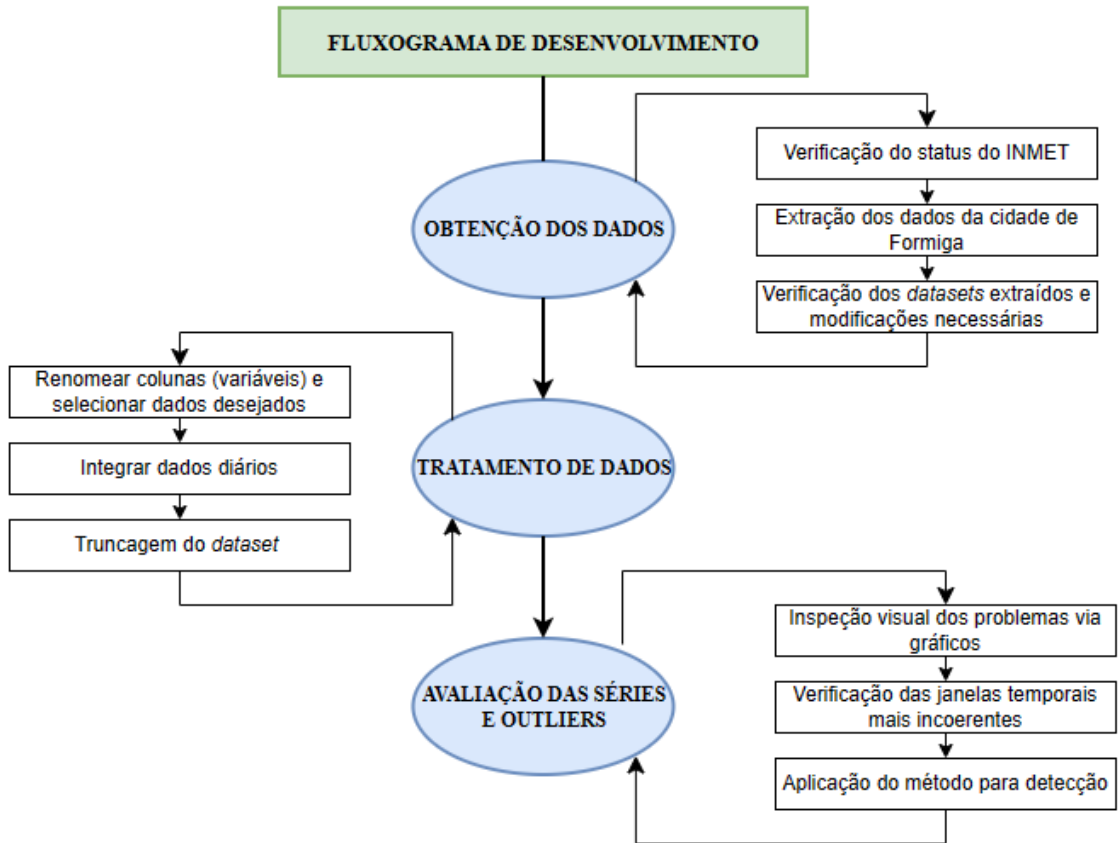
Fonte: Solar (2023).

Os painéis fotovoltaicos (1), que convertem a energia solar em corrente contínua, são conectados à entrada do inversor de frequência (2) que, por sua vez, entrega tensão alternada à unidade consumidora (3) com frequência e valor eficaz nominais adequados aos valores de frequência e tensão de fornecimento da concessionária. A partir de então, esta energia elétrica gerada por meio desta fonte de energia renovável está pronta para o consumo. É muito importante salientar que esta energia gerada, antes de consumida, é direcionada a um medidor bidirecional (4), situado no padrão de entrada da instalação, que realiza a medição do montante de energia gerado por esta unidade consumidora e do montante que a mesma utiliza da rede elétrica convencional (caso a demanda da instalação seja superior ao montante gerado pela usina). Consequentemente, possibilita para a unidade consumidora a diminuição no valor da conta de energia elétrica, consolidado por meio da entrega do excedente de geração para a rede de distribuição (5). Por fim, esta unidade deixa de ser apenas um consumidor e se torna um consumidor-gerador (*prosumer*).

### 3 METODOLOGIA

O processo desenvolvido para realização do trabalho, de forma resumida, está apresentado no fluxograma da Figura 3.

Figura 3 – Fluxograma de desenvolvimento geral.



Fonte: Elaborado pelo autor (2023).

Inicialmente é feita a obtenção automática dos dados do INMET, onde é verificado o *status* da disponibilidade dos dados do *site*. Após a verificação, um dicionário com os *links* para *download* é criado, onde todos os anos de todas as cidades ficam disponíveis. Para este trabalho, foram extraídos os dados referentes ao município de Formiga. Foi necessária uma configuração dos dados devido a uma alteração na formatação dos dados ocorrida entre os anos 2000 e 2023.

Em seguida, dá-se início ao tratamento dos dados. Os nomes das colunas foram transformados em variáveis para facilitar o acesso aos dados. O INMET disponibiliza os dados de forma horária e, para facilitar a visualização e o trabalho com estes dados, foram integrados para dados diários. A base de dados também foi dividida entre dados para treinamento e teste, para futura utilização em modelos de previsão.

Para detecção dos *outliers*, foi feita uma inspeção visual dos gráficos anuais e assim detectadas as janelas temporais mais coerentes. Com isso, foi aplicado o método de detecção

dos *outliers*, que utiliza uma versão modificada do algoritmo KNN (k-vizinhos mais próximos) (KNORR; NG; TUCAKOV, 2000).

### 3.1 Aquisição dos dados

Para a aquisição dos dados meteorológicos necessários para a previsão da energia solar, foi utilizada a base de dados históricos anuais do Instituto Nacional de Meteorologia (INMET), disponibilizada em seu portal <https://portal.inmet.gov.br/dadoshistoricos> (INMET, 2023).

Para cada ano, um arquivo *.zip* é disponibilizado com uma série de tabelas *.csv* das cidades com estações meteorológicas cadastradas. Na Figura 4 pode ser visto um exemplo dos dados meteorológicos do ano de 2022, enquanto que na Figura 5 são mostrados dados do município de Formiga, que será a região estudada. Dados de precipitação, pressão atmosférica, radiação global, temperatura, umidade e velocidade do vento estão presentes no arquivo disponibilizado, organizados estes em colunas, enquanto que nas linhas se referem aos intervalos de tempo medidos.

Com o intuito de automatizar a aquisição dos dados foi desenvolvida uma ferramenta na linguagem de programação Julia que verifica a disponibilidade de séries anuais climatológicas no recurso `/dadoshistoricos` do portal. Este método foi implementado na função `statusINMET()` do pacote *SolarPowerForecastingIFMG*, seu retorno consiste de um dicionário com a estrutura:

```
"20XX"=> "https://portal.inmet.gov.br/uploads/dadoshistoricos/20XX.zip",
```

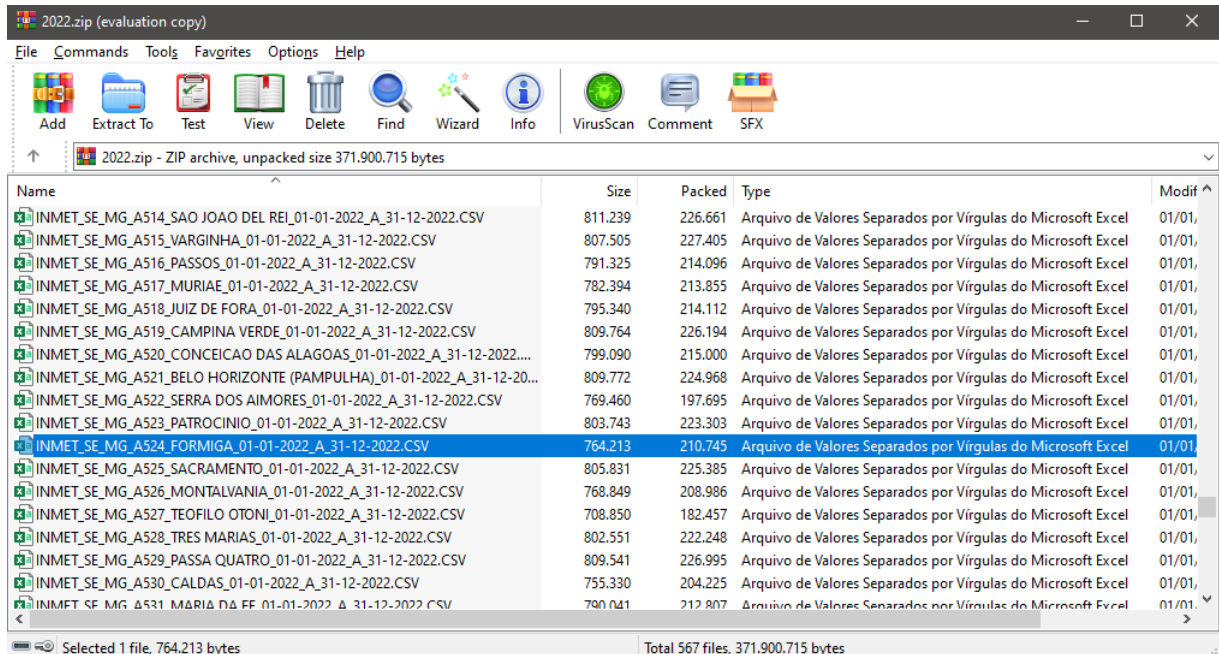
onde as chaves são os anos disponíveis na base do formato de um inteiro de 32 *bits* e os valores uma *string* contendo o *link* para *download* dos arquivos *.zip* descritos no início desta sessão.

Após a instalação, ativação do ambiente ou atualização do pacote em desenvolvimento (*SolarPowerForecasting*), o procedimento de verificação do *status* da plataforma do INMET pode ser realizada.

O início do processo de raspagem de dados do INMET consiste requisição *http* ao portal no recurso `/dadoshistoricos`, para extração dos links e textos dos *datasets* de cada série anual, para que possam ser realizados os *downloads*. Este processo, presente no código `RaspagemDeDadosINMET.jl`, acontece de forma automatizada por meio da função `statusINMET()`. A variável `fonte_dados`, é do tipo dicionário e recebe desta função os *links* para acesso às tabelas *.csv* de cada ano. Por meio da função `obter_dados` acontece a extração de dados propriamente dita, que tem como argumentos:

- `fonte_dados`: dicionário com chaves do tipo `Int16`, que contém o ano e o *link* de *download* de determinada série histórica anual;
- `idades`: vetor de *strings* que contém a lista de cidades a serem estudadas;

Figura 4 – Arquivos disponíveis para ano de 2022



Fonte: Elaborado pelo autor (2023).

Figura 5 – Dados disponíveis para ano de 2022 para cidade de Formiga

Data	Hora UTC	PRECIPITAO	PRESSAO	PRESSÃO	RADIACAO	TEMPERA	TEMPERA	TEMPERA	TEMPERA	TEMPERA	TEMPERA	UMIDADE	UMIDADE	VENTO, DI	VENTO, R	VENTO, VELOCIDADE HORA
0000 UTC	0,8	912,6	912,6	911,8	19,7	18,8	20,1	19,6	18,8	18,6	94	92	94	332	11,9	6,7
0100 UTC	1,2	912,8	912,9	912,6	19,8	19	19,9	19,6	19	18,7	95	94	95	350	11,9	5,3
0200 UTC	1	912,6	912,9	912,6	19,6	18,6	20	19,6	19,1	18,4	95	92	94	347	12,9	6,3
0300 UTC	0,8	912,3	912,6	912,3	18,3	17,2	19,6	18,2	18,6	17,1	95	91	93	351	14,3	6,7
0400 UTC	2	912,3	912,5	912,2	17,8	16,6	18,3	17,8	17,2	16,5	93	92	93	8	14,2	4,5
0500 UTC	4,2	911,8	912,3	911,8	17,7	16,9	17,8	17,5	17	16,6	96	93	95	16	9,8	2,9
0600 UTC	2,4	911,7	911,8	911,6	17,9	16,8	18	17,6	16,9	16,7	95	93	93	4	10,1	5,2
0700 UTC	1,8	912,2	912,2	911,7	17,9	16,9	18,1	17,8	17	16,7	94	92	94	357	12,3	4,5
0800 UTC	1,2	912,1	912,3	912,1	0	17,6	16,4	18	17,5	17	94	93	93	27	9,1	3,3
0900 UTC	0,8	912,9	912,9	912,1	54,3	18,3	17,4	18,5	17,5	17,4	95	89	95	351	8,9	3,8
1000 UTC	0	913,5	913,5	912,8	258,9	19,3	17,6	19,3	18,3	17,7	95	90	90	5	9,4	4,4
1100 UTC	0	914	914	913,5	981,2	20,6	18	20,7	19,3	18,3	90	84	85	2	10,9	4,7
1200 UTC	0	914,4	914,4	914	1530,8	21,2	17,9	21,5	20,6	18,5	86	81	81	335	11,9	7,4
1300 UTC	0	914,2	914,4	914,2	1569,3	22,4	18,2	22,4	21,3	18,6	83	76	77	339	11,9	6,4
1400 UTC	0	913,7	914,2	913,7	2704,3	24,5	18,6	24,5	22,4	18,7	77	68	70	350	12,6	6,4
1500 UTC	0	913	913,7	913	3138	24	17,8	24,7	23,1	18,5	73	66	68	341	13,7	7,2
1600 UTC	0	912,3	913	912,2	2433,7	24,1	17,7	24,8	23,4	18,6	72	67	68	356	13,7	5,7
1700 UTC	0	911,8	912,3	911,8	2712,1	25	18,7	25,6	23,9	19,1	71	65	68	3	11,6	4,9

Fonte: Elaborado pelo autor (2023).

- *interv\_temp*: um vetor de inteiros que contém uma lista dos anos a serem estudados ou um *UnitRang* com este intervalo de estudo.

Como o exemplo abaixo, tem-se a requisição dos dados para a cidade de Formiga no ano de 2010.

```
dados = obter_dados(fonte_dados, ["FORMIGA"], [2010]);
```

Na Figura 6 é exibida a variável dados. Nela é mostrada parte da tabela, que se contém as mesmas informações descritas para a Figura 5.

Figura 6 – Dados requeridos de Formiga no ano de 2010.

Row	DATA (YYYY-MM-DD) Dates.Date	HORA (UTC) Dates.Time	PRECIPITA\xc7\xc30 TOTAL, HOR\xc... Float64	PRES ... Floa ...
1	2010-01-01	00:00:00		0.0
2	2010-01-01	01:00:00		0.0
3	2010-01-01	02:00:00		0.0
4	2010-01-01	03:00:00		0.0
5	2010-01-01	04:00:00		0.0
6	2010-01-01	05:00:00		0.0
7	2010-01-01	06:00:00		0.0
8	2010-01-01	07:00:00		0.0
9	2010-01-01	08:00:00		0.0
10	2010-01-01	09:00:00		0.0
11	2010-01-01	10:00:00		0.0
⋮	⋮	⋮	⋮	⋮
8751	2010-12-31	14:00:00		0.6
8752	2010-12-31	15:00:00		2.8
8753	2010-12-31	16:00:00		0.4
8754	2010-12-31	17:00:00		0.8
8755	2010-12-31	18:00:00		1.4
8756	2010-12-31	19:00:00		2.8
8757	2010-12-31	20:00:00		0.0
8758	2010-12-31	21:00:00		15.6
8759	2010-12-31	22:00:00		0.6
8760	2010-12-31	23:00:00		0.2

Fonte: Elaborado pelo autor (2023).

Existe uma falha de codificação na fonte de dados que desconfigura a rotulação dos dados, como, por exemplo, na terceira coluna da Figura 6, onde se tem "PRECIPTA\xc7\xc30". Para a resolução do problema, foi realizada uma inspeção na lista de nomes das colunas utilizando a função names no objeto do tipo dataset pertencente à mutable struct EstruturaDeCaptura.

Para resolução desse problema de forma automatizada, foi criado um dicionário com os índices das colunas a serem modificadas e os novos nomes, seguido de um ajuste iterativo utilizando a função ajst\_colnames!().

A função obter\_dados possibilita uma aquisição múltipla de dados, tanto para obter informações para mais de uma cidade, quanto para distintos intervalos de tempo.

Para acessar as cidades desejadas para o estudo, basta declará-las no segundo parâmetro da função (desde que seja uma cidade disponível no portal do INMET). Já a aquisição de dados

para múltiplos períodos pode ser feita de duas formas. A primeira simplesmente declarando o terceiro parâmetro da função `obter_dados` como um vetor com os anos desejados para estudo. A segunda forma, permitida para situações onde o conjunto de dados se dá para um período sequencial de anos, a determinação do período pode ser realizada via `UnitRange`, que permite que o parâmetro seja escrito de uma forma reduzida, na forma `"interv_temp = 2010:2014"`, por exemplo.

### 3.2 Tratamento dos dados

O tratamento dos dados obtidos na seção anterior é importante com o objetivo de retirar colunas que não são necessárias no modelo, ou seja, selecionar apenas os parâmetros desejados para implementação. Os dados são de radiação, temperatura e pluviosidade. Os nomes das colunas de cada um desses parâmetros foram alterados, substituindo-os do tipo *string* para o tipo *symbol*, a fim de facilitar o acesso aos dados temporais de cada parâmetro.

Os dados faltantes, os chamados *missing values*, foram substituídos por valores nulos, para que a base de dados tenha ao menos um valor onde não existia dado algum. Com isso, os dados que naturalmente são horários vindos da base do INMET, foram integrados para dados diários, para facilitar o trabalho e visualização.

Para que o tratamento listado acima seja realizado de forma automatizada, foi implementada a função `treat_data!()`, que recebe como argumento a variável `dados`, responsável por armazenar os dados obtidos anteriormente. Na Figura 7 é mostrado o novo *dataset* redimensionado e reconfigurado com informações necessárias. Pode ser notado a mudança na nomenclatura das colunas, ocasionado pelo fato de não serem representadas mais por *strings* mas por símbolos da linguagem Julia, o que facilita o referenciamento aos dados.

Figura 7 – Novo *dataset* redimensionado.

	ADRAIN	ADSOLPW	DTMAX_C	DTMIN_C	DATE
1	0.6	24396.6	28.2	19.1	2019-01-01
2	0.0	27677.2	30.9	17.8	2019-01-02
3	0.0	21536.9	30.1	20.2	2019-01-03
4	7.8	8292.9	25.3	19.5	2019-01-04
5	0.0	15369.1	26.3	19.4	2019-01-05

Fonte: Elaborado pelo autor (2023).

Para demonstração de um exemplo prático do tratamento destes dados foi feita a obtenção de um gráfico para avaliação de radiação global anual. Com as linhas de código descritas na Figura

8, é possível obter um gráfico demonstrando visualmente o comportamento anual da grandeza em 2019.

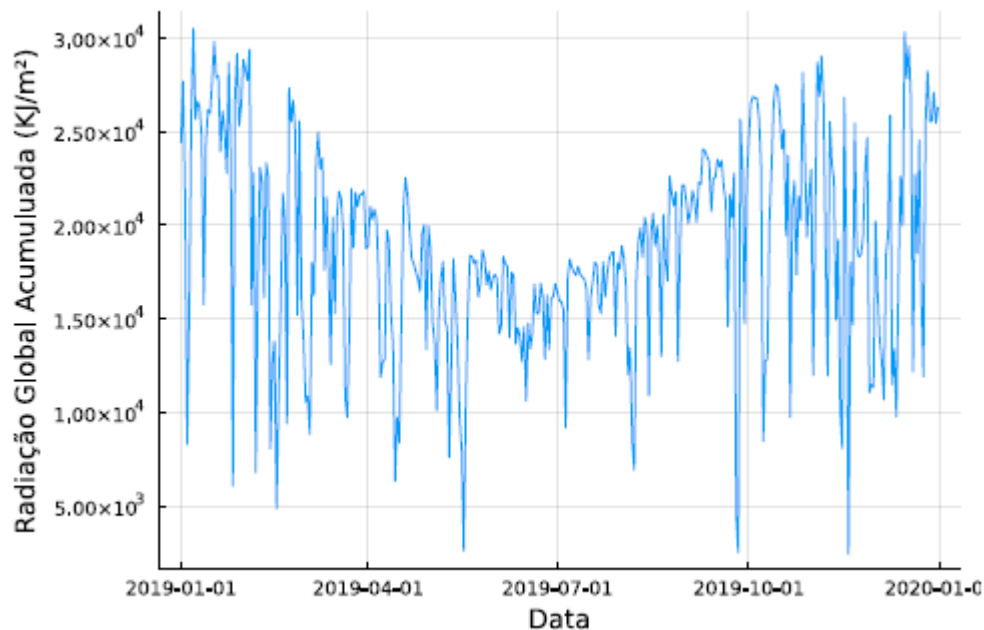
Figura 8 – Codificação para avaliação do comportamento de radiação solar acumulada anual.

```
1 begin
2   plot(dados.serie[1].dataset.DATE, dados.serie[1].dataset.ADSOLPW, label=false)
3   xlabel!("Data")
4   ylabel!("Radiação Global Acumulada (KJ/m²)")
5 end
```

Fonte: Elaborado pelo autor (2023).

A função `plot` recebe como parâmetros a informações de data (referenciada pelo símbolo `DATE`) e radiação solar global acumulada (referenciada pelo símbolo `ADSOLPW`). Na Figura 9 o gráfico obtido é apresentado.

Figura 9 – Gráfico para avaliação do comportamento de radiação solar acumulada anual



Fonte: Elaborado pelo autor (2023).

Outro exemplo é a correlação entre os dados de máxima temperatura diária e de radiação global acumulada no dia, conforme a codificação da Figura 10.

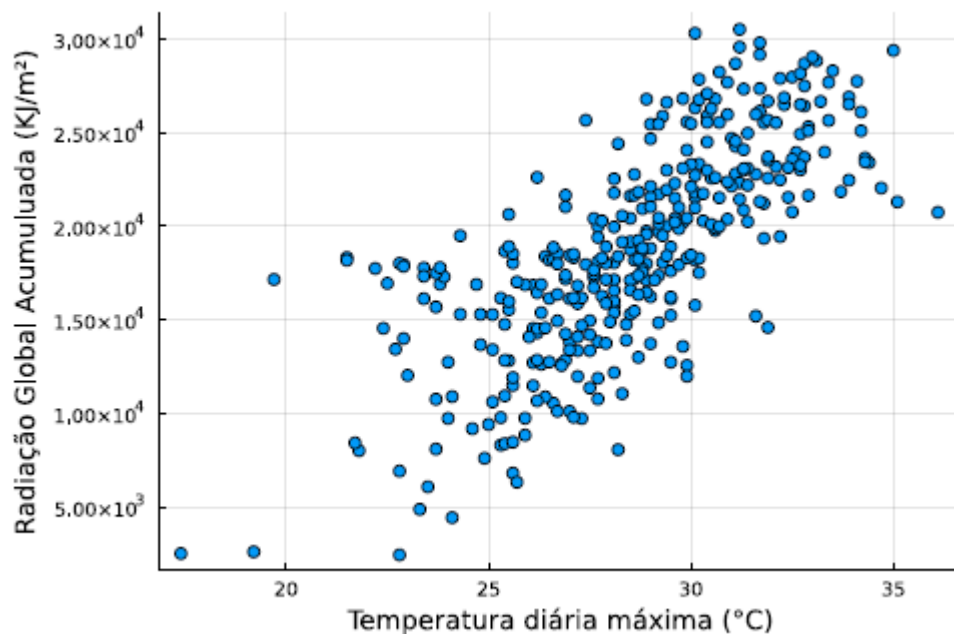
A função `scatter` correlaciona e plota o gráfico com os parâmetros de máxima temperatura diária (referenciada pelo símbolo `DTMAX_C`) e radiação solar global acumulada. Na Figura 11 o gráfico obtido é apresentado.

Figura 10 – Codificação para avaliação da correlação entre máxima temperatura diária e radiação solar acumulada

```
1 begin
2 scatter(dados.serie[1].dataset.DTMAX_C, dados.serie[1].dataset.ADSOLPW,
3 label=false)
4 xlabel!("Temperatura diária máxima (°C)")
5 ylabel!("Radiação Global Acumulada (KJ/m²)")
6 end
```

Fonte: Elaborado pelo autor (2023).

Figura 11 – Gráfico da correlação entre máxima temperatura diária e radiação solar acumulada



Fonte: Elaborado pelo autor (2023).

### 3.3 Truncagem do *dataset*

Para que seja feita uma futura previsão, por exemplo utilizando uma rede neural, para que seja realizado o seu treinamento é necessário que seja feita a divisão dos dados em *train set*, e *test set* para a validação. A função `split_df` do módulo `TruncagemDeDados` faz esse trabalho de forma automatizada.

### 3.4 Avaliação de séries e *outliers*

Um dos problemas encontrados no tratamento de dados reais, no caso, os dados provenientes das estações meteorológicas automáticas do município de Formiga, é a presença de *outliers*.

Estes são os dados que têm comportamentos destoantes dos demais.

Para visualização desse fenômeno, foi analisado um conjunto de dados referentes aos anos de 2010 até 2023, conforme codificação da Figura 12.

Figura 12 – Obtenção dos *datasets* de 2010 a 2023 para avaliação dos *outliers*

```
1 begin
2   fonte_dados = statusINMET();
3   dados = obter_dados(fonte_dados, ["FORMIGA"], 2010:2023);
4   ajst_colnames!(dados);
5   treat_data!(dados);
6 end
```

Fonte: Elaborado pelo autor (2023).

Com todos estes *datasets* em mãos, um para cada ano, foram plotados todos os gráficos de cada uma das grandezas estudadas para uma inspeção inicial dos *outliers* e para verificação da melhor janela temporal a ser utilizada.

A janela temporal definida foi concatenada, ou seja, os dados que anteriormente estavam definidos ano a ano, agora fazem parte de uma única série.

Para a identificação propriamente dita dos *outliers*, foi utilizada a técnica utilizada por Knorr, Ng e Tucakov (2000), que é uma versão do algoritmo KNN, descrito na Seção 2.4 deste trabalho.

O processo inicia-se com a instalação dos pacotes necessários, neste caso, os pacotes MLJ e OutlierDetection. O pacote OutlierDetection, disponibiliza uma máquina de predição que cria o modelo de predição, este por sua vez é passado ao método de treinamento que efetivamente identifica os pontos. Com os pontos identificados, a base de dados agora conta com uma coluna que diz se cada ponto, ou seja, cada dado é considerado pelo algoritmo um dado normal ou um *outlier*.

Com os *outliers* identificados foi realizada a filtragem dos dados confiáveis para exportação de um arquivo .csv.

## 4 RESULTADOS E DISCUSSÕES

### 4.1 Tratamento de problemas de codificação de caracteres

Na Figura 13 é mostrado o *dataset* com o nome das colunas desconfiguradas. Após a implementação da função, na Figura 14 são apresentadas a nova lista dos nomes das colunas com a correta configuração.

Figura 13 – Lista de nomes das colunas desconfigurada

```
20-element Vector{String}:
"DATA (YYYY-MM-DD)"
"HORA (UTC)"
"PRECIPITA\xc7\xc3O TOTAL, HOR\xc1RIO (mm)"
"PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO, HORARIA (mB)"
"PRESS\xc3O ATMOSFERICA MAX.NA HORA ANT. (AUT) (mB)"
"PRESS\xc3O ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)"
"RADIACAO GLOBAL (KJ/m\xb2)"
"TEMPERATURA DO AR - BULBO SECO, HORARIA (\xb0C)"
"TEMPERATURA DO PONTO DE ORVALHO (\xb0C)"
"TEMPERATURA M\xc1XIMA NA HORA ANT. (AUT) (\xb0C)"
"TEMPERATURA M\xcDNIMA NA HORA ANT. (AUT) (\xb0C)"
"TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (\xb0C)"
"TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (\xb0C)"
"UMIDADE REL. MAX. NA HORA ANT. (AUT) (%)"
"UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)"
"UMIDADE RELATIVA DO AR, HORARIA (%)"
"VENTO, DIRE\xc7\xc3O HORARIA (gr) (\xb0 (gr))"
"VENTO, RAJADA MAXIMA (m/s)"
"VENTO, VELOCIDADE HORARIA (m/s)"
"Column20"
```

Fonte: Elaborado pelo autor (2023).

Com isso, foi possível visualizar melhor a informações fornecidas pelo INMET, onde tem-se:

- Data no formato ANO-MÊS-DIA;
- Hora (UTC - *Coordinated Universal Time*);
- Precipitação total, por hora (mm);
- Pressão atmosférica máxima na hora anterior (mB);
- Pressão atmosférica mínima na hora anterior (mB);
- Radiação global (kJ/m<sup>2</sup>);

Figura 14 – Lista de nomes das colunas configurada.

```

20-element Vector{String}:
"DATA (YYYY-MM-DD)"
"HORA (UTC)"
"PRECIPITACAO TOTAL, HORARIA (mm)"
"PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO, HORARIA (mB)"
"PRESSAO ATMOSFERICA MAX.NA HORA ANT.(AUT) (mB)"
"PRESSAO ATMOSFERICA MIN.NA HORA ANT.(AUT) (mB)"
"RADIACAO GLOBAL (KJ/m²)"
"TEMPERATURA DO AR - BULBO SECO, HORARIA (°C)"
"TEMPERATURA DO PONTO DE ORVALHO (°C)"
"TEMPERATURA MAXIMA NA HORA ANT.(AUT) (°C)"
"TEMPERATURA MINIMA NA HORA ANT.(AUT) (°C)"
"TEMPERATURA ORVALHO MAX. NA HORA ANT. (AUT) (°C)"
"TEMPERATURA ORVALHO MIN. NA HORA ANT. (AUT) (°C)"
"UMIDADE REL. MAX. NA HORA ANT. (AUT) (%)"
"UMIDADE REL. MIN. NA HORA ANT. (AUT) (%)"
"UMIDADE RELATIVA DO AR, HORARIA (%)"
"VENTO, DIRECAO HORARIA (gr) (°(gr))"
"VENTO, RAJADA MAXIMA (m/s)"
"VENTO, VELOCIDADE HORARIA (m/s)"
"Column20"

```

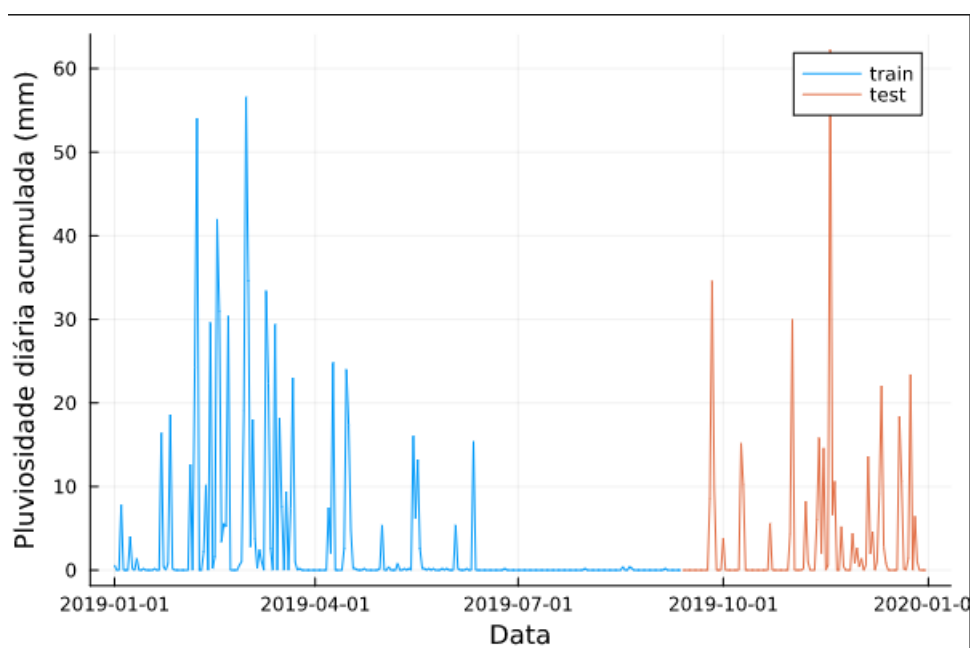
Fonte: Elaborado pelo autor (2023).

- Temperatura do ar, bulbo seco, por hora (°C);
- Temperatura do ponto de orvalho (°C);
- Temperatura máxima na hora anterior (°C);
- Temperatura mínima na hora anterior (°C);
- Temperatura de orvalho máxima na hora anterior (°C);
- Temperatura de orvalho mínima na hora anterior (°C);
- Umidade relativa máxima na hora anterior (%);
- Umidade relativa mínima na hora anterior (%);
- Umidade relativa do ar, por hora (%);
- Direção do vento, por hora (°);
- Rajada máxima de vento (m/s);
- Velocidade do vento, por hora (m/s).

## 4.2 Dados truncados para futuro treinamento de redes neurais

Nos gráficos a seguir pode ser verificada a truncagem para treino e teste dos dados referentes à: pluviosidade diária acumulada, na Figura 15, radiação global diária acumulada, na Figura 16, e máximas e mínimas temperaturas diárias, na Figura 17 .

Figura 15 – Gráfico truncagem dados de pluviosidade diária acumulada



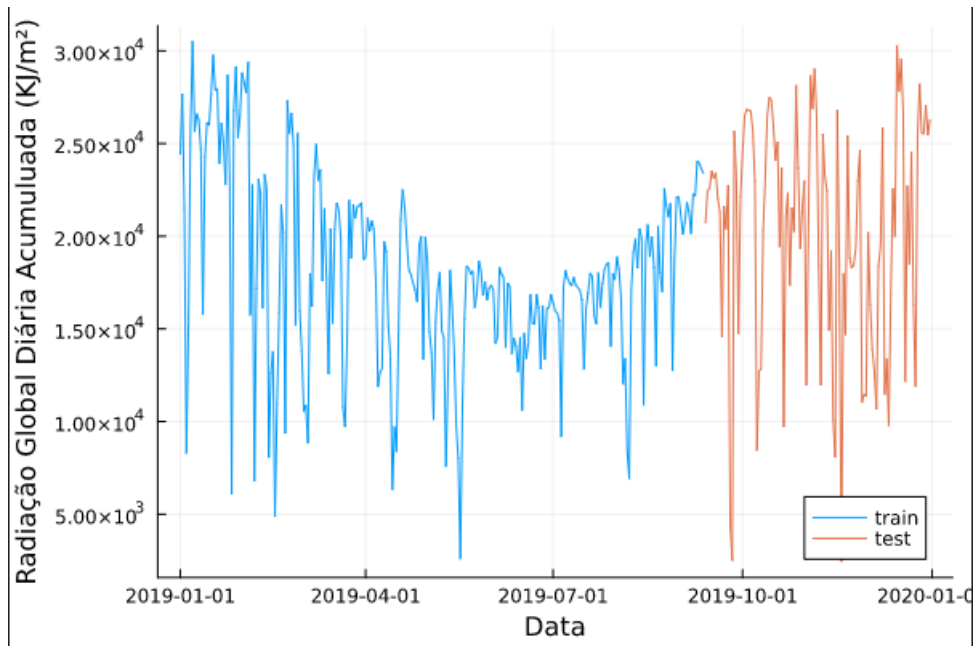
Fonte: Elaborado pelo autor (2023).

## 4.3 Detecção dos *outliers*

Para inspeção inicial dos *outliers*, é possível observar nos gráficos apresentados nas Figuras 18, 19, 20 e 21, que mostram respectivamente os dados de radiação, pluviosidade, temperatura máxima diária e temperatura mínima diária para a cidade de Formiga entre os anos de 2010 e 2023, que os dados à partir de 2019 apresentam um comportamento mais estável, sendo que neste tempo foram padronizadas as unidades dentro do banco de dados do INMET, e estes serão utilizados como padrão. Importante frisar que foi realizada uma inspeção visual dos gráficos para determinar a janela temporal melhor a ser trabalhada.

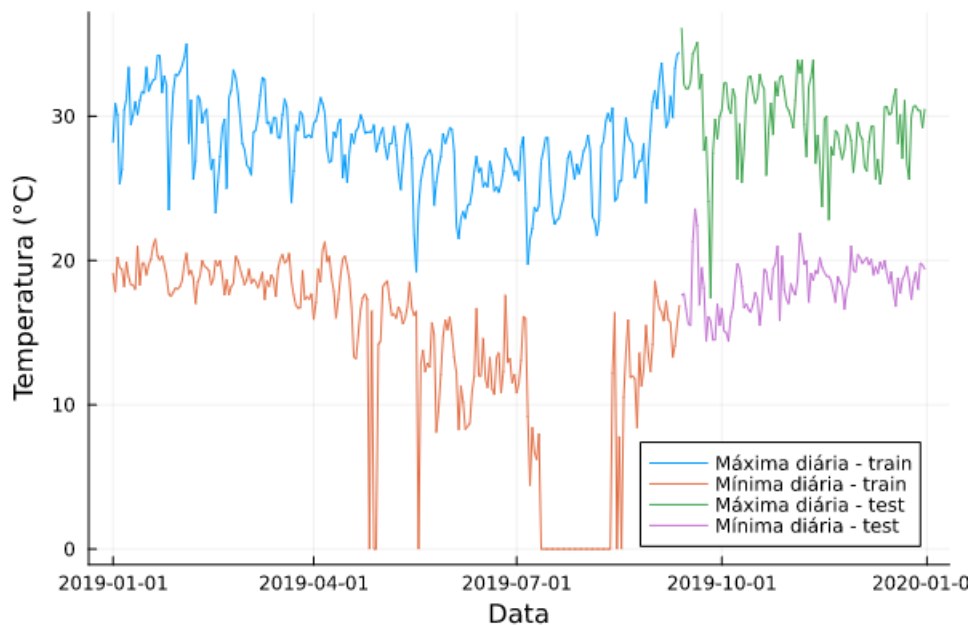
Com a janela temporal definida, os dados foram concatenados, fazendo parte de um mesmo conjunto de dados. A Figura 22 mostra os dados das grandezas de radiação, pluviosidade, temperatura máxima e temperatura mínima de 2019 até 2023, agora em um único *dataset*, não mais separados por ano. Para visualizar graficamente o processo, a Figura 23 mostram os dados em conjunto de radiação de 2019 a 2023.

Figura 16 – Gráfico truncagem dados de radiação global diária acumulada



Fonte: Elaborado pelo autor (2023).

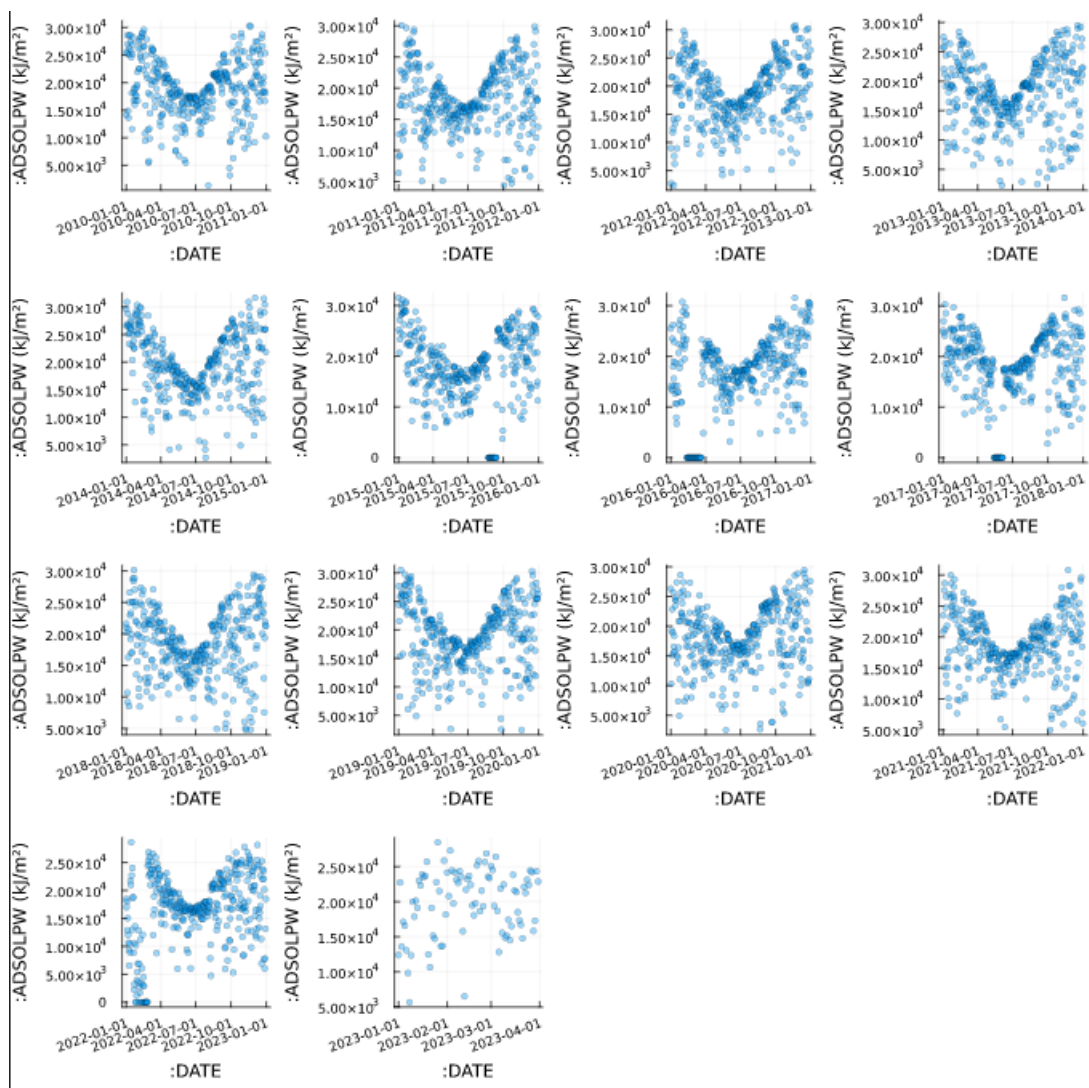
Figura 17 – Gráfico truncagem dados de amplitude térmica



Fonte: Elaborado pelo autor (2023).

Com os dados obtidos e a aplicação do algoritmo de detecção de *outliers*, na Figura 24 é mostrado o novo *dataset*. Nela agora os conjuntos de dados de radiação entre 2019 e 2023 são rotulados entre "normal" ou "outlier", identificados por uma nova coluna chamada "OUTLIER\_YN".

Figura 18 – Gráfico dos dados de radiação solar global de 2010 até 2023



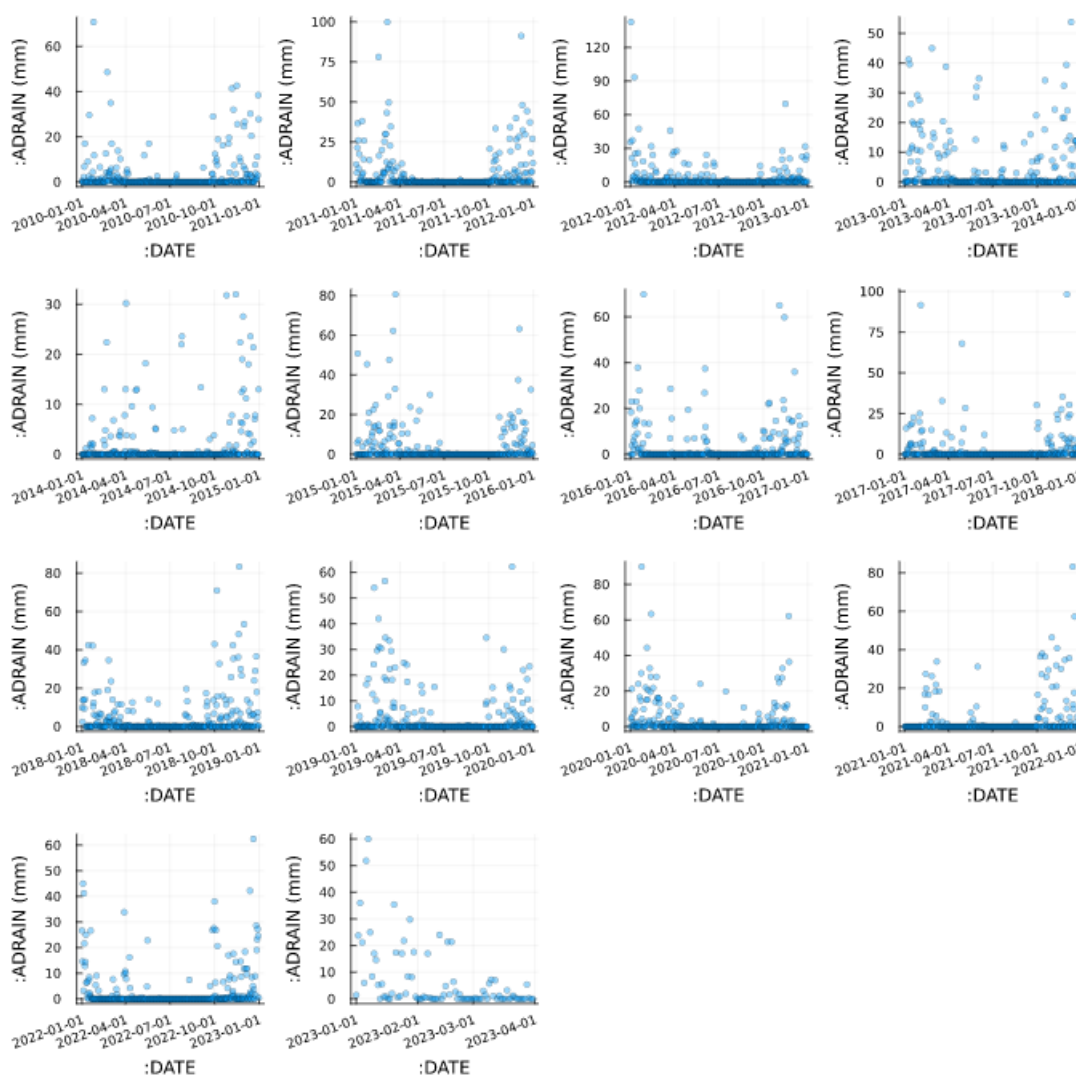
Fonte: Elaborado pelo autor (2023).

Para que esse efeito seja observado de forma gráfica, na Figura 25 é mostrado o gráfico dos dados de radiação, onde são identificados os *outliers* detectados em vermelho. É possível observar os pontos mais destoantes, os que, de acordo com o algoritmo aplicado são os *outliers*. Pode parecer estranho alguns pontos que parecem coerentes visualmente, porém o gráfico está em escala *semi-log*, o que pode fazer a distância visual ser "enganosa".

Com a filtragem dos dados, afim de obter um conjunto de dados mais confiáveis, a tabela exportada com dados de temperatura diária máxima e temperatura diária mínima para março de 2023 é mostrada na Figura 26. Nela é mostrada parte da tabela *.csv* que foi exportada, esta que pode ser manipulada tanto com mudança entre os parâmetros desejados, janela temporal desejadas e também visualizar apenas os dados considerados "normais".

A partir dos resultados obtidos, algumas discussões e análises podem ser feitas. É visto

Figura 19 – Gráfico dos dados de pluviosidade de 2010 até 2023



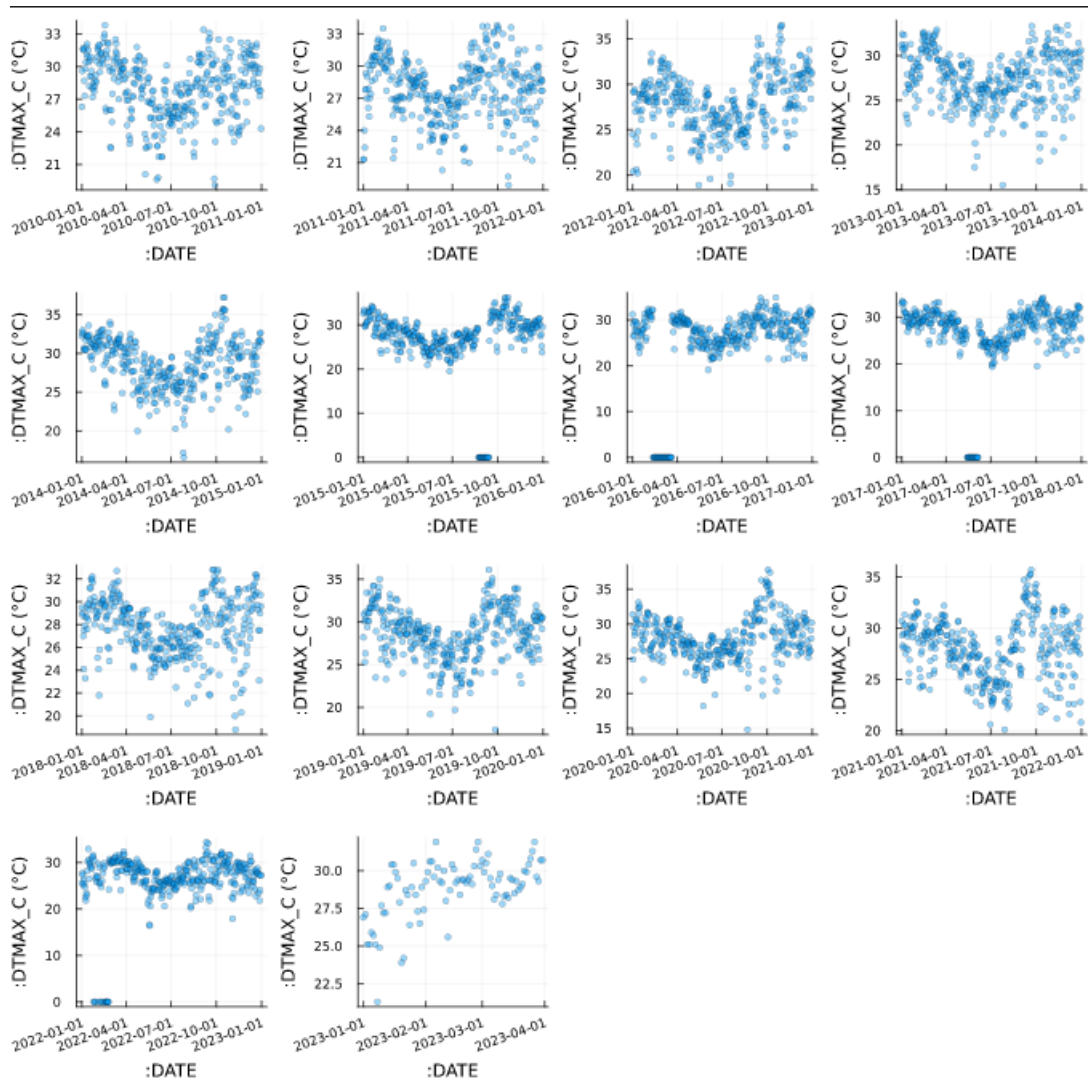
Fonte: Elaborado pelo autor (2023).

que a base de dados do INMET, se obtida de forma direta, carrega consigo vários erros que, possivelmente, vem de problemas no equipamento de medição, principalmente antes do ano de 2019, e também de codificação, como foi visto na nomenclatura das colunas.

Na visualização gráfica dos dados de todas as grandezas, seja de radiação global acumulada, pluviosidade e temperaturas máximas e mínimas, é possível ver o comportamento sazonal de todas elas. Os dados são coerentes neste sentido pois mostram o comportamento da região de Formiga, que entre setembro e março possuem maior quantidade de radiação global, maiores índices de pluviosidade e temperaturas mais altas, enquanto que entre abril e agosto estes números são bem menores. Esta característica pode ser útil para uma futura implementação de métodos de predição.

Os mostrados na Figura 25 e 26, é possível verificar que o método aplicado, ao fazer

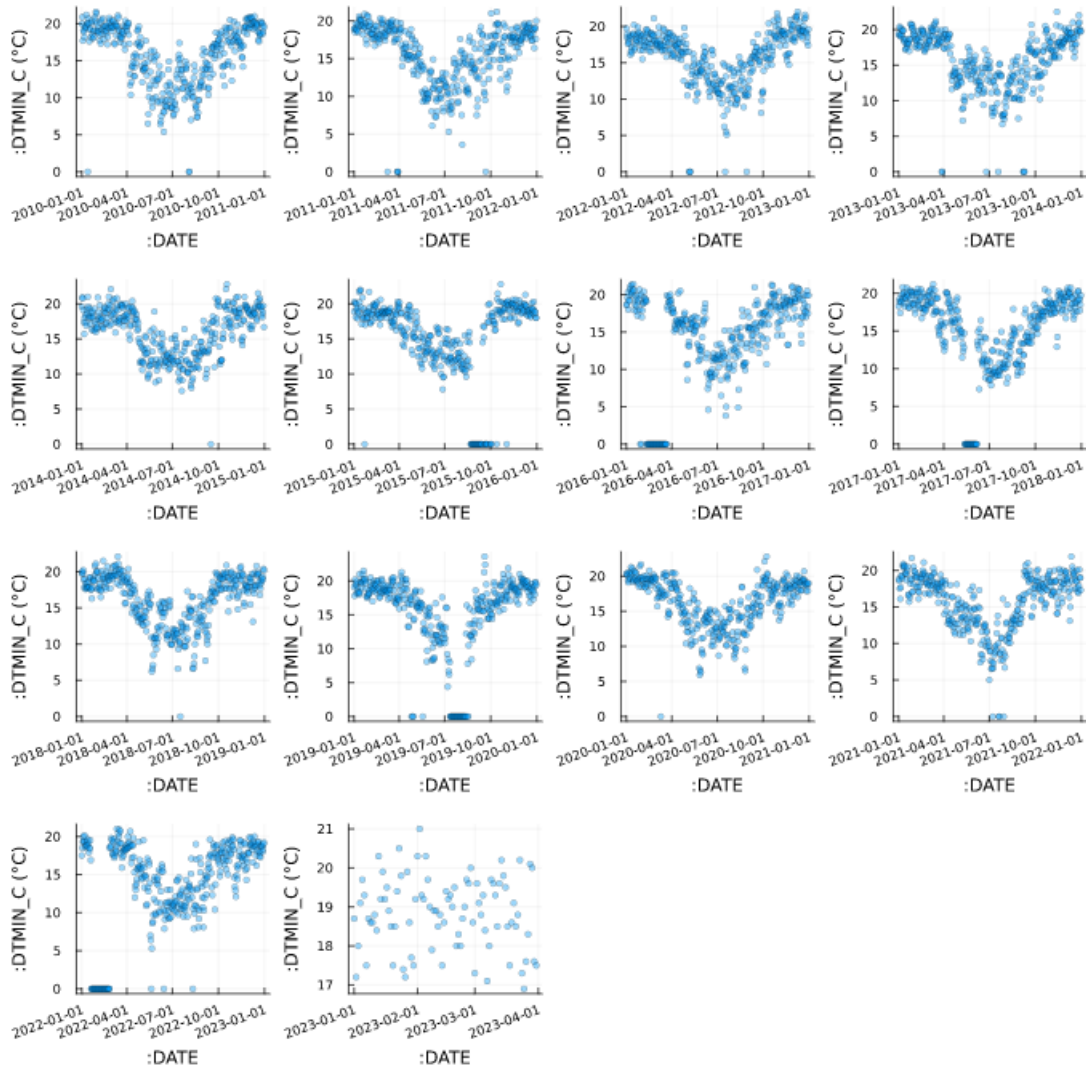
Figura 20 – Gráfico dos dados de temperatura máxima diária de 2010 até 2023



Fonte: Elaborado pelo autor (2023).

a detecção dos *outliers*, possibilitou a criação de um novo conjunto de dados, uma nova base, mais limpa e confiável, o qual era o objetivo deste trabalho. Com esta nova base, é possível uma aplicação de um método de previsão desta série para que seja avaliada a provável potência solar desta área, com os dados meteorológicos disponibilizados pelo INMET.

Figura 21 – Gráfico dos dados de temperatura mínima diária de 2010 até 2023



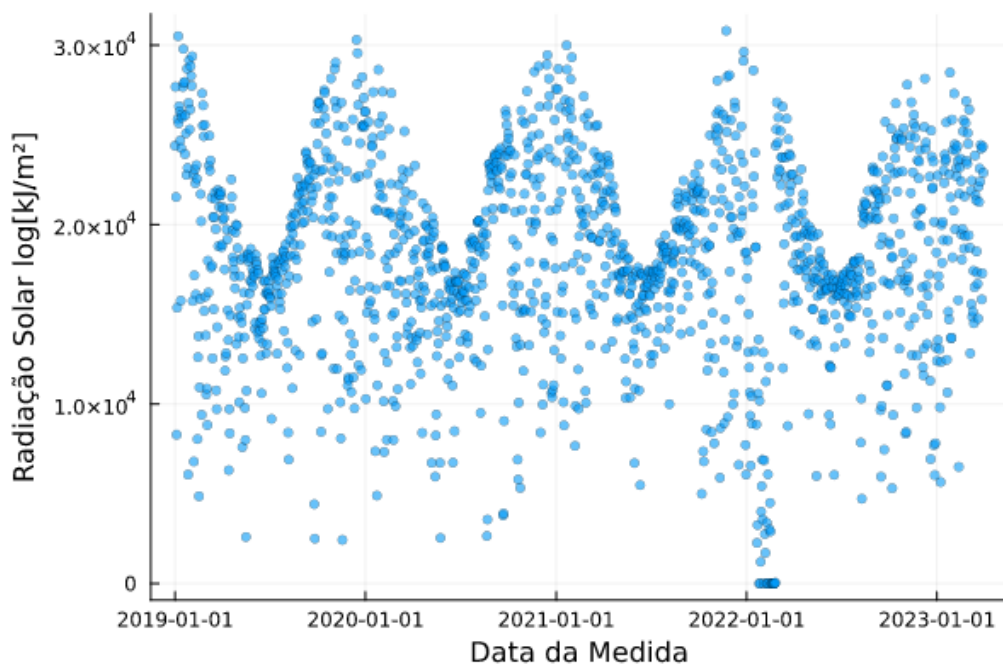
Fonte: Elaborado pelo autor (2023).

Figura 22 – Dados concatenados de 2019 a 2023

	ADRAIN	ADSOLPW	DTMAX_C	DTMIN_C	DATE
1	0.6	24396.6	28.2	19.1	2019-01-01
2	0.0	27677.2	30.9	17.8	2019-01-02
3	0.0	21536.9	30.1	20.2	2019-01-03
4	7.8	8292.9	25.3	19.5	2019-01-04
5	0.0	15369.1	26.3	19.4	2019-01-05
6	0.0	25874.5	30.4	18.2	2019-01-06
7	0.0	30501.9	31.2	19.9	2019-01-07
8	4.0	25638.5	33.4	18.9	2019-01-08
9	0.2	26599.4	29.4	18.3	2019-01-09
10	0.0	26307.4	30.1	18.3	2019-01-10
⋮	more				
1537	0.0	22891.5	30.7	17.5	2023-03-31

Fonte: Elaborado pelo autor (2023).

Figura 23 – Gráfico de dados de radiação concatenados de 2019 a 2023



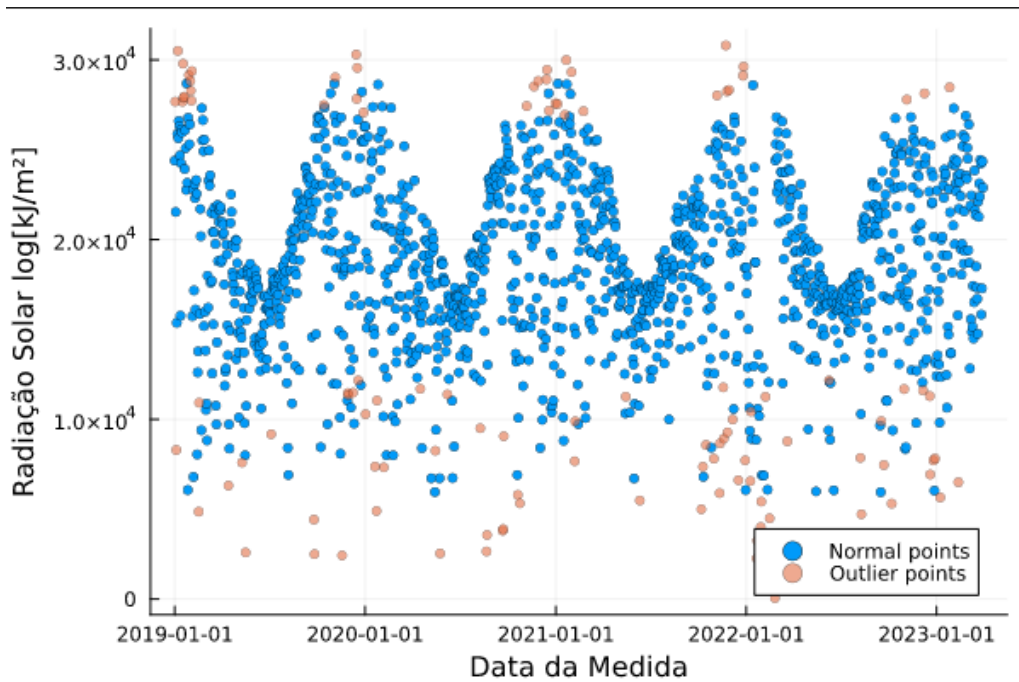
Fonte: Elaborado pelo autor (2023).

Figura 24 – Novo dataset com os outliers identificados

	ADRAIN	ADSOLPW	DTMAX_C	DTMIN_C	DATE	OUTLIER_YN
1	0.6	24396.6	28.2	19.1	2019-01-01	"normal"
2	0.0	27677.2	30.9	17.8	2019-01-02	"outlier"
3	0.0	21536.9	30.1	20.2	2019-01-03	"normal"
4	7.8	8292.9	25.3	19.5	2019-01-04	"outlier"
5	0.0	15369.1	26.3	19.4	2019-01-05	"normal"
6	0.0	25874.5	30.4	18.2	2019-01-06	"normal"
7	0.0	30501.9	31.2	19.9	2019-01-07	"outlier"
8	4.0	25638.5	33.4	18.9	2019-01-08	"normal"
9	0.2	26599.4	29.4	18.3	2019-01-09	"normal"
10	0.0	26307.4	30.1	18.3	2019-01-10	"normal"
⋮	more					
50	5.6	21699.8	29.2	19.0	2019-02-19	"normal"

Fonte: Elaborado pelo autor (2023).

Figura 25 – Gráfico de dados de radiação com os outliers identificados



Fonte: Elaborado pelo autor (2023).

Figura 26 – Tabela de dados exportada com os dados filtrados

DTMAX_C	DTMIN_C	DATE	OUTLIER_YN
30.7	17.3	01/03/2023	normal
29.9	19.6	02/03/2023	normal
30.5	19.1	03/03/2023	normal
31.1	18.8	04/03/2023	normal
29.5	19.3	05/03/2023	normal
29.0	18.4	06/03/2023	normal
28.1	17.1	07/03/2023	normal
28.4	18.0	08/03/2023	normal
29.4	19.7	09/03/2023	normal
29.8	19.6	10/03/2023	normal
27.8	19.3	11/03/2023	normal
28.3	18.5	12/03/2023	normal
28.4	19.6	13/03/2023	normal
28.2	20.2	14/03/2023	normal
29.5	19.8	15/03/2023	normal
29.4	19.5	16/03/2023	normal
28.5	18.5	17/03/2023	normal
29.3	17.5	18/03/2023	normal
28.7	18.6	19/03/2023	normal
28.9	19.1	20/03/2023	normal
29.1	18.5	21/03/2023	normal
30.0	18.8	22/03/2023	normal
29.9	20.2	23/03/2023	normal
30.4	17.3	24/03/2023	normal
30.8	16.9	25/03/2023	normal
31.3	17.6	26/03/2023	normal
31.9	18.3	27/03/2023	normal
29.6	20.1	28/03/2023	normal
29.3	20.0	29/03/2023	normal
30.7	17.6	30/03/2023	normal
30.7	17.5	31/03/2023	normal

Fonte: Elaborado pelo autor (2023).

## 5 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de uma ferramenta computacional que, de forma automática, extrai dados do INMET (Instituto Nacional de Meteorologia), faz o tratamento destes dados e por fim, faz uma verificação destes dados detectando os *outliers*.

Para o desenvolvimento da ferramenta foi utilizada a linguagem Julia, por ser uma linguagem nova que combina programação dinâmica de alto nível, alto desempenho e eficiência. Adicionalmente, esta linguagem de programação conta com uma grande quantidade de pacotes e bibliotecas que facilitam o desenvolvimento e permitem maiores possibilidades de aplicação.

Para a verificação dos *outliers*, foi aplicada uma versão modificada do algoritmo KNN (k-vizinhos mais próximos) desenvolvido por Knorr, Ng e Tucakov (2000), que verifica a distância do ponto com seus vizinhos, e dependendo desta distância, é rotulado ou não como um *outlier*.

Com a metodologia utilizada os objetivos foram alcançados, sendo que foi possível detectar e visualizar os *outliers*, os dados destoantes dos demais. Com isso, uma base limpa foi criada, capaz de alimentar ferramentas de previsão e levar a elas informações mais confiáveis.

O pacote desenvolvido está disponível no repositório do *Github* do GESESC (Grupo de Estudos em Sistemas Energéticos e Simulação Computacional) (GESESC, 2023).

### 5.1 Trabalhos futuros

Com a análise e tratamento dos dados devidamente realizadas, propõe-se algumas possibilidades de trabalhos futuros:

- aplicação de outros métodos de detecção de *outliers* (como os descritos na Seção 2.4) e comparação entre eles;
- aplicação de métodos de previsão de séries temporais, como por exemplo o ARMA, ARIMA, redes neurais recorrentes, entre outros, descritos na Seção 2.5 deste trabalho, para com isso, fazer uma previsão da potência solar fotovoltaica a ser gerada na região de Formiga.

## REFERÊNCIAS

- ANTONIO, C. Healthcare in smart cities-a review of data mining with julia vs python vs r. 2019.
- AXILLUS, V. **Comparing Julia and Python: An investigation of the performance on image processing with deep neural networks and classification.** 2020.
- AZEVEDO, L. L. d. **Uma análise exploratória de dados aplicado ao mercado financeiro.** Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2019.
- BACHER, P.; MADSEN, H.; NIELSEN, H. A. Online short-term solar power forecasting. **Solar Energy**, Elsevier BV, v. 83, n. 10, p. 1772–1783, out. 2009. Disponível em: <<https://doi.org/10.1016/j.solener.2009.05.016>>.
- BASTOS, I. G. P. *et al.* Previsão de geração fotovoltaica a partir de dados meteorológicos utilizando rede lstm. In: **Congresso Brasileiro de Energia Solar-CBENS.** [S.l.: s.n.], 2020.
- BEZANSON, J.; EDELMAN, A.; KARPINSKI, S.; SHAH, V. B. **Julia: A Fresh Approach to Numerical Computing.** arXiv, 2014. Disponível em: <<https://arxiv.org/abs/1411.1607>>.
- BEZANSON, J.; KARPINSKI, S.; SHAH, V. B.; EDELMAN, A. **Julia: A Fast Dynamic Language for Technical Computing.** arXiv, 2012. Disponível em: <<https://arxiv.org/abs/1209.5145>>.
- \_\_\_\_\_. **Julia: A Fast Dynamic Language for Technical Computing.** arXiv, 2012. Disponível em: <<https://arxiv.org/abs/1209.5145>>.
- CALÇADA, R. d. O. Design of steel: a risc-v core. 2020.
- CAMPOS, D. R. D. de. **Análise quantitativa do impacto da ausência de dados.** Tese (Doutorado), 2021. Disponível em: <<https://doi.org/10.11606/d.3.2021.tde-31082021-111051>>.
- CAMPOS, G. O. **Estudo, avaliação e comparação de técnicas de detecção não supervisionada de outliers.** Tese (Doutorado), 2015. Disponível em: <<https://doi.org/10.11606/d.55.2015.tde-04082015-084412>>.
- ENERGIA, M. D. M. E. **ENERGIA RENOVÁVEL CHEGA À QUASE 50ELÉTRICA BRASILEIRA.** 2022. Disponível em: <<https://www.gov.br/pt-br/noticias/energia-minerais-e-combustiveis/2021/08/energia-renovavel-chega-a-quase-50-da-matriz-eletrica-brasileira>>.
- FIEKER, C.; HART, W.; HOFMANN, T.; JOHANSSON, F. Nemo/hecke: Computer algebra and number theory packages for the julia programming language. arXiv, 2017. Disponível em: <<https://arxiv.org/abs/1705.06134>>.
- FREITAS, I. W. S. d. *et al.* Um estudo comparativo de técnicas de detecção de outliers no contexto de classificação de dados. Universidade Federal Rural do Semi-Árido, 2019.
- GESESC. **Desenvolvimento preliminar de um pacote para previsão automática de potência solar baseado em dados históricos do INMET.** 2023. Disponível em: <<https://github.com/GESESC/SolarPowerForecastingIFMG.jl>>.
- GREEN, M. A.; EMERY, K.; KING, D. L.; HISIKAWA, Y.; WARTA, W. Solar cell efficiency tables (version 27). **Progress in Photovoltaics**, Chichester, West Sussex, England: Wiley, c1993-, v. 14, n. 1, p. 45–52, 2006.

INMET. **DADOS HISTÓRICOS ANUAIS**. 2023. Disponível em: <<https://portal.inmet.gov.br/dadoshistoricos>>.

JESUS, D. F. de *et al.* Predição solar e eólica utilizando redes com neurônios de histerese 12p para aplicações em smart grids. In: **Anais do Simpósio Brasileiro de Sistemas Elétricos 2020**. sbabra, 2020. Disponível em: <<https://doi.org/10.48011/sbse.v1i1.2340>>.

JULIAHUB. **Dynamic and Diverse**. 2023. Disponível em: <<https://juliahub.com/case-studies/>>.

KNORR, E. M.; NG, R. T.; TUCAKOV, V. Distance-based outliers: algorithms and applications. **The VLDB Journal**, Springer, v. 8, n. 3, p. 237–253, 2000.

LAIA, I. W. G. de. Aplicação da linguagem julia em ciência de dados. 2021.

LEWANDOWSKI, D. G. Análise do impacto de geração distribuída na rede de distribuição. Universidade Federal de Santa Maria, 2023.

LIANG, M. **RegTools: A Julia Package for Assisting Regression Analysis**. Tese (Doutorado) — UCLA, 2015.

MAGALHAES, J. W. de S.; AVALOS, J. B.; SANTOS, L. J. A. dos. Dossiê linguagem julia. 2017.

MARYNYCH, T. O.; NAZARENKO, L. D.; KHOMENKO, N. H. Comparative analysis of univariate time series modeling and forecasting techniques for short-term unstable data. « », 2017.

NETO, A. O. Análise do impacto da geração distribuída em um sistema de distribuição. estudo de caso: Alimentador do setor de embaixadas sul, Brasília-df. **Universidade de Brasília-UnB. Brasília**, 2016.

OLIVEIRA, C. M. d. Uma análise exploratória de dados e a construção de modelos preditivos de machine learning no cálculo de risco de subscrição de uma seguradora. 2023.

\_\_\_\_\_. Uma análise exploratória de dados e a construção de modelos preditivos de machine learning no cálculo de risco de subscrição de uma seguradora. 2023.

OLSON, D. L.; DELEN, D. **Advanced data mining techniques**. [S.l.]: Springer Science & Business Media, 2008.

PAULA, M. G. d. Aprendizagem cruzada para previsão de séries temporais univariadas. 2023.

RAI, A.; SHRIVASTAVA, A.; JANA, K. C. Differential attention net: Multi-directed differential attention based hybrid deep learning model for solar power forecasting. **Energy**, Elsevier BV, v. 263, p. 125746, jan. 2023. Disponível em: <<https://doi.org/10.1016/j.energy.2022.125746>>.

RAMOS, A. S. Previsões de séries temporais combinando modelos arma e redes neurais artificiais. Universidade Federal de Pernambuco, 2010.

REINALDO, P. M.; DUPCZAK, B. S.; NETO, E. A. C. A. Impact assessment of photovoltaic distributed generation on celesc low-voltage network. **Eletrônica de Potência**, Associação Brasileira de Eletronica de Potencia SOBRAEP, v. 25, n. 3, p. 316–325, ago. 2020. Disponível em: <<https://doi.org/10.18618/rep.2020.3.0013>>.

RIO, P. **A Modelagem Univariada por Box & Jenkins**. 2023. Disponível em: <[https://www.maxwell.vrac.puc-rio.br/8161/8161\\_4.PDF](https://www.maxwell.vrac.puc-rio.br/8161/8161_4.PDF)>.

SEDAI, A. *et al.* Performance analysis of statistical, machine learning and deep learning models in long-term forecasting of solar power production. **Forecasting**, MDPI AG, v. 5, n. 1, p. 256–284, fev. 2023. Disponível em: <<https://doi.org/10.3390/forecast5010014>>.

SIE. **Capacidade Instalada de Geração Elétrica**. 2023. Disponível em: <<https://www.mme.gov.br/SIEBRASIL/consultas/reporte-dato42-jerarquizado.aspx?oc=30181&or=30182&ss=2&v=1>>.

SOLAR, M. **O que é um Sistema Fotovoltaico?** 2023. Disponível em: <<https://www.mmsolar.com.br/single-post/2018/11/28/o-que-%C3%A9-um-sistema-fotovoltaico>>.

SUNDERLAND, K. M. *et al.* The utility of multivariate outlier detection techniques for data quality evaluation in large studies: an application within the ondri project. **BMC medical research methodology**, BioMed Central, v. 19, n. 1, p. 1–16, 2019.

TAVARES, L. A. MATRIZ ELÉTRICA BRASILEIRA e AS TENDÊNCIAS FUTURAS. **RECIMA21 - Revista Científica Multidisciplinar - ISSN 2675-6218**, RECIMA21 - Revista Científica Multidisciplinar, v. 4, n. 5, p. e453135, maio 2023. Disponível em: <<https://doi.org/10.47820/recima21.v4i5.3135>>.

TUSHAR, D. T. *et al.* Binary priority outlier classifier based outlier elimination. **Turkish Journal of Computer and Mathematics Education (TURCOMAT)**, Auricle Technologies, Pvt., Ltd., v. 12, n. 3, p. 4261–4266, abr. 2021. Disponível em: <<https://doi.org/10.17762/turcomat.v12i3.1717>>.

VIEIRA, F. C. Análise exploratória de dados: limpeza, manipulação e pré-processamento aplicado a dataset de perfil de atendimento nas unidades de saúde da cidade de Curitiba. Universidade Tecnológica Federal do Paraná, 2021.