

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA

LANA PAULA PEREIRA GOMES

**PREVISIBILIDADE DE GERAÇÃO SOLAR NO CONTEXTO DE HIBRIDIZAÇÃO
DE FONTES PARA ANÁLISE DE COMPLEMENTARIDADE**

UBERLÂNDIA - MG

2023

LANA PAULA PEREIRA GOMES

**PREVISIBILIDADE DE GERAÇÃO SOLAR NO CONTEXTO DE HIBRIDIZAÇÃO
DE FONTES PARA ANÁLISE DE COMPLEMENTARIDADE**

Trabalho apresentado a Universidade Federal
de Uberlândia – UFU, Campus Santa Mônica,
como requisito para obtenção do título de
Bacharel em Engenharia Elétrica.

Área de concentração:

Orientador: Prof. Dr. Ivan Nunes Santos

Coorientadora: Ms. Camila de Oliveira Dias

UBERLÂNDIA - MG

2023

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

G633 2023	<p>Gomes, Lana Paula Pereira, 1999- PREVISIBILIDADE DE GERAÇÃO SOLAR NO CONTEXTO DE HIBRIDIZAÇÃO DE FONTES PARA ANÁLISE DE COMPLEMENTARIDADE [recurso eletrônico] / Lana Paula Pereira Gomes. - 2023.</p> <p>Orientador: Ivan Nunes Santos. Coorientadora: Camila de Oliveira Dias. Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Uberlândia, Graduação em Engenharia Elétrica. Modo de acesso: Internet. Inclui bibliografia.</p> <p>1. Engenharia elétrica. I. Santos, Ivan Nunes, 1979-, (Orient.). II. Dias, Camila de Oliveira, 1993-, (Coorient.). III. Universidade Federal de Uberlândia. Graduação em Engenharia Elétrica. IV. Título.</p> <p style="text-align: right;">CDU: 621.3</p>
--------------	---

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

LANA PAULA PEREIRA GOMES

**PREVISIBILIDADE DE GERAÇÃO SOLAR NO CONTEXTO DE HIBRIDIZAÇÃO
DE FONTES PARA ANÁLISE DE COMPLEMENTARIDADE**

Trabalho apresentado a Universidade Federal de Uberlândia – UFU, Campus Santa Mônica, como requisito para obtenção do título de Bacharel em Engenharia Elétrica.

Uberlândia, 16 de junho de 2023

Banca Examinadora:

Ivan Nunes Santos – Doutor (UFU)

Camila de Oliveira Dias – Mestre (UFU)

Ábner César Peres Pacheco – Mestre (UFU)

Dedico aos meus pais, Simone e Claudio.

AGRADECIMENTOS

Agradeço primeiramente a Deus por sempre colocar as pessoas certas em meu caminho e por ele ter me ajudado a ingressar na universidade dos meus sonhos, além de me auxiliar na conclusão desta última etapa.

Agradeço aos meus pais, que sempre me incentivaram, me guiaram e me apoiaram em todas as minhas decisões acadêmicas, sendo, sem dúvida, meus maiores incentivadores e minha base.

Agradeço ao meu irmão por ser um exemplo de engenheiro e por ter me dado o maior presente que eu poderia receber: a Laura. Tê-la em minha vida tornou tudo mais leve e contribuiu para o meu desempenho no Trabalho de Conclusão de Curso.

Agradeço ao Hander por sempre proferir palavras positivas, acreditando constantemente em meu potencial.

Agradeço aos meus amigos Gladson e Paula, que estiverem ao meu lado durante toda essa jornada, dando-me forças nos momentos mais difíceis da graduação. Foi maravilhoso estudar com vocês.

Agradeço, também, ao Professor Doutor Ivan Nunes e à minha coorientadora Camila pela orientação durante esse trabalho.

“Tudo o que um sonho precisa para ser realizado é alguém que acredite que ele possa ser realizado.”

(SHINYASHIKI, 2005)

RESUMO

No intuito de combinar duas fontes renováveis para contribuir com a redução de gases carbônicos, contribuindo para uma matriz energética mais sustentável, foram conduzidas pesquisas sobre a integração entre usina fotovoltaica e usina hidrelétrica. Nesse contexto, torna-se necessário aprofundar-se nos modelos estatísticos e algoritmos de previsibilidade para identificar o modelo mais adequado na realização das previsões de irradiância, uma vez que para a geração solar não se tem clareza a respeito dos modelos estabelecidos para auxiliar nesse processo. Essa etapa de desenvolvimento é fundamental para viabilizar futuros estudos de viabilidade de complementaridade entre as fontes. Para tanto, selecionou-se os seguintes métodos: *Autoregressive Integrated Moving Average*, *Autoregressive Integrated Moving Average with Exogenous Variables*, *Long Short Term Memory*, *Gated Recurrent Unit* e *Convolutional Neural Network*, os quais foram, nesta pesquisa, implementados utilizando a linguagem Python no Google Colaboratory. Em seguida, traçou-se o perfil da curva da média diária para cada modelo, comparando-a com os valores dos dados reais. Para obter os dados reais desta comparação, foram feitas as previsões para o ano de 2022, e os dados correspondentes foram armazenados em um DataFrame separado para a realização de métricas de avaliação. Com base nas métricas estabelecidas, observou-se que os modelos estatísticos *Autoregressive Integrated Moving Average* e *Autoregressive Integrated Moving Average with Exogenous Variables* tiveram maior complexibilidade na implementação devido aos parâmetros necessários e à sazonalidade presente nos dados. Assim, os resultados foram piores em comparação aos demais algoritmos de redes neurais. Dentre os algoritmos de previsibilidade avaliados, o *Convolutional Neural Network* apresentou os melhores resultados.

Palavras-chave: complementariedade; modelos estatísticos; algoritmos de previsibilidade; Python; métricas de avaliação.

ABSTRACT

In order to combine two renewable sources to contribute to the reduction of carbon dioxide, contributing to a more sustainable energy matrix, research was conducted on the integration between photovoltaic plant and hydroelectric plant. In this context, it was necessary to delve deeper into statistical models and predictability algorithms to identify the most appropriate model for making irradiance predictions, since for solar generation there are no established models to assist in this process. This step was essential to enable future studies on the feasibility of complementarity between the sources. After selecting the *Autoregressive Integrated Moving Average*, *Autoregressive Integrated Moving Average with Exogenous Variables*, *Long Short Term Memory*, *Gated Recurrent Unit* and *Convolutional Neural Network methods*, they were implemented using the Python language in COLABORATORY. Then, the profile of the curve of the daily average for each model was traced, comparing it with the values of the real data. To obtain actual data for comparison, predictions were made for the year 2022, and the corresponding data was stored in a separate DataFrame for performing evaluation metrics. Based on the established metrics, it was observed that the statistical models *Autoregressive Integrated Moving Average* and *Autoregressive Integrated Moving Average with Exogenous Variables* presented greater complexity in the implementation due to the parameters to be defined and the seasonality present in the data, with that, the results were inferior in compared to other neural network algorithms. Among these predictability algorithms, *the Convolutional Neural Network* showed the best results.

Keywords: complementarity; statistical models; predictability algorithms; Python; evaluation metrics.

LISTA DE ILUSTRAÇÕES

Figura 1 – Matriz de Energia Elétrica 2023	12
Figura 2 - Capacidade Instalada de Energia Solar entre Abril de 2018 e Março de 2023	13
Figura 3 - Comparação da Capacidade Instalada de Energia Hídrica de 2018 com 2022	13
Figura 4 - Benefícios da Descarbonização	14
Figura 5 - Usina Híbrida.....	19
Figura 6 - Usina Associada.....	19
Figura 7 - Energia solar fotovoltaica flutuante em Rosana, São Paulo	21
Figura 8 - Esquema de montagem dos painéis solares sobre a água	21
Figura 11 - Usina hidrelétrica de armazenamento	22
Figura 12 - Usina hidrelétrica a fio d'água.....	23
Figura 13 - Usina hidrelétrica reversível	23
Figura 14 - Prazos dos modelos computacionais.	25
Figura 15 - Ciclo iterativo do modelo ARIMA	29
Figura 16 - Como criar um algoritmo de previsibilidade.	30
Figura 17- Célula de memória.	32
Figura 18 - Arquitetura de uma Rede Neural GRU.....	36
Figura 19 - Camadas da Rede Neural Convolucional.	36
Figura 20 - Conjunto de dados	41
Figura 21 - Dados de Treino 2017 a 2021	43
Figura 22 - Dados para comparação dos resultados	43
Figura 23 - Média diária dos dados de treino	44
Figura 24 - Média diária dos dados de teste	44
Figura 25 - Decomposição sazonal usando o método da decomposição aditiva.....	46
Figura 26 - Gráfico de Autocorrelação.....	47
Figura 27 - Gráfico de Autocorrelação Parcial.....	48
Figura 28 - Média diária do modelo ARIMA sazonal.....	49
Figura 29 - Média diária do modelo ARIMAX sazonal	52
Figura 30 - Média diária do algoritmo LSTM.....	56
Figura 31 - Média diária do algoritmo GRU	59
Figura 32 - Média diária do algoritmo CNN	63
Figura 33 - Comparação da média diária dos modelos e algoritmos.....	65

LISTA DE TABELAS

Tabela 1 -	Métricas de avaliação.....	64
Tabela 2 -	Amostras fora do limite de 15%.....	65

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
AR	<i>Autoregressive</i>
ARIMA	<i>Autoregressive Integrated Moving Average</i>
ARIMAX	<i>Autoregressive Integrated Moving Average with Exogenous Variables</i>
ARMA	<i>Autoregressive Moving Average</i>
CCEE	Câmara de Comercialização de Energia Elétrica
CEPEL	Centro de Pesquisas de Energia Elétrica
CMO	Custo Marginal de Operação
CNN	<i>Convolutional Neural Network</i>
EPE	Empresa de Pesquisa Energética
GRU	<i>Gated Recurrent Unit</i>
I	<i>Integrated</i>
INMET	Instituto Nacional de Meteorologia
LSTM	<i>Long Short Term Memory</i>
MA	<i>Moving Average</i>
MAE	Erro Médio Absoluto
MME	Ministério de Minas e Energia
MQO	Mínimos Quadrados Ordinários
ONS	Operador Nacional do Sistema Elétrico
PLD	Preço de Liquidação das Diferenças
R ²	Coefficiente de Determinação
SONDA	Sistema de Organização Nacional de Dados Ambientais
UFU	Universidade Federal de Uberlândia
UHE	Usina Hidrelétrica
UHR	Usina Hidrelétrica Reversível
URI	Uniform Resource Identifier

SUMÁRIO

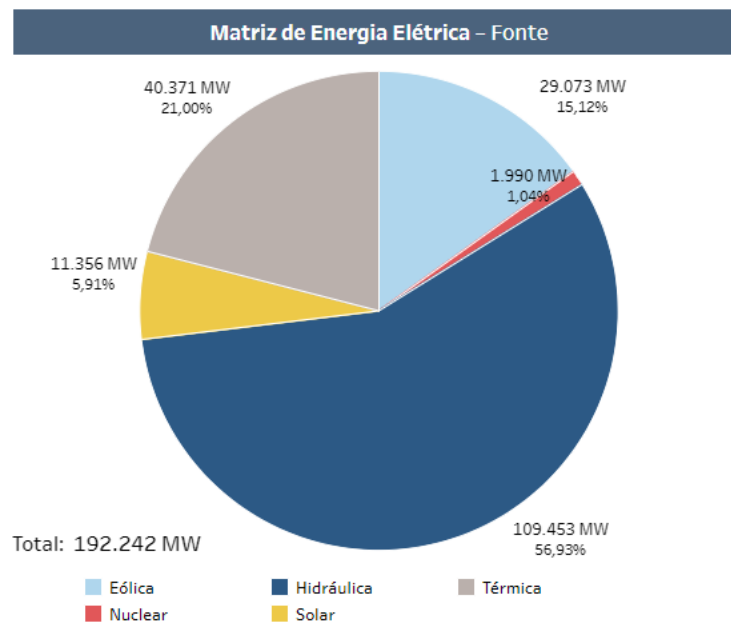
1	INTRODUÇÃO.....	12
1.1	MOTIVAÇÕES	15
1.2	OBJETIVO GERAL	15
1.3	ESTRUTURA DO TRABALHO.....	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	SISTEMA HÍBRIDO	18
2.1.1	REGULAÇÃO	19
2.1.2	DESAFIOS	20
2.1.3	COMBINAÇÃO ENTRE HIDROELÉTRICA E FOTOVOLTAICA.....	20
2.2	POTENCIAL DE ENERGIA HIDRÁULICA NO PAÍS.....	22
2.3	POTENCIAL DE ENERGIA SOLAR NO PAÍS	25
2.4	MODELO DE PREVISÃO ESTATÍSTICO.....	26
2.4.1	ARMA.....	26
2.4.2	ARIMA	28
2.4.3	ARIMAX	29
2.5	ALGORITMOS PARA PREVISIBILIDADE	30
2.6.1	LSTM.....	31
2.6.2	GRU	34
2.6.3	CNN	36
2.7	MÉTRICAS DE AVALIAÇÃO	37
2.7.1	ERRO MÉDIO ABSOLUTO – MAE.....	37
2.7.2	COEFICIENTE DE DETERMINAÇÃO – R ²	37
3	DESENVOLVIMENTO DOS ESTUDOS DE DESEMPENHO	39
3.1	SELEÇÃO DOS DADOS.....	39
3.2	ANÁLISE E TRATAMENTO DOS DADOS	40
3.3	APLICAÇÃO DOS MODELOS E ALGORITMOS DE PREVISIBILIDADE	40
3.3.1	ARIMA	45
3.3.2	ARIMAX	51
3.3.3	LSTM.....	53
3.3.4	GRU	57
3.3.5	CNN	60
3.4	COMPARAÇÃO DOS RESULTADOS	64

4 CONCLUSÃO	67
REFERÊNCIAS	69

1 INTRODUÇÃO

O conjunto das fontes de energia empregadas no território nacional é chamada matriz energética brasileira, a qual é considerada uma das mais diversificadas e renováveis do mundo, por ser majoritariamente renovável, tendo como destaque a fonte hídrica que corresponde a 56,93% da potência instalada na matriz elétrica, totalizando aproximadamente 80% da oferta interna de eletricidade no Brasil de fontes renováveis, assim como pode ser visto na Figura 1 [1].

Figura 1 – Matriz de Energia Elétrica 2023



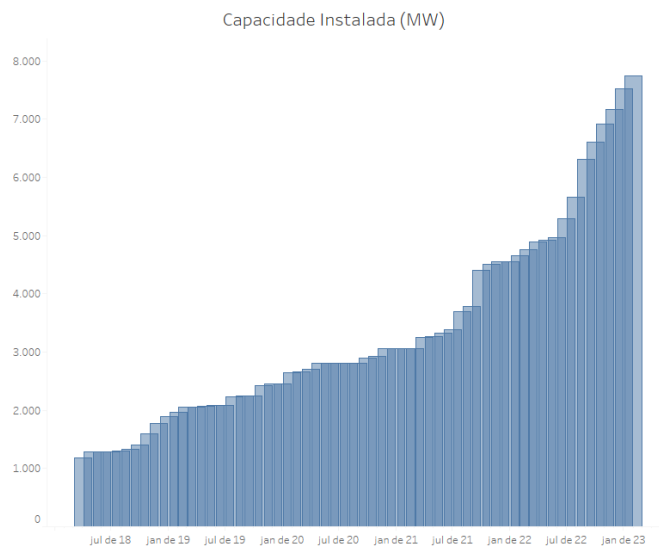
Fonte: [2]

Segundo o Ministério de Minas e Energia (MME), houve uma expansão de 2.746,5 MW da capacidade instalada de geração de energia elétrica nos primeiros três meses de 2023 e esse valor equivale ao dobro ao se comparar com o mesmo período em 2022 [3]. De acordo com a Agência Nacional de Energia Elétrica (ANEEL), 82 novas usinas entraram em funcionamento até o final de março de 2023, sendo a maioria composta por usinas fotovoltaicas e parques eólicos, o que indica que o país está se desenvolvendo na ampliação de fontes renováveis [4]. Além disso a geração distribuída teve um aumento de 18,5% na potência instalada no primeiro trimestre de 2023 [5].

De acordo com dados disponibilizados pelo Operador Nacional do Sistema Elétrico (ONS), é possível observar, por meio das Figuras 2 e 3, que a capacidade instalada de geração

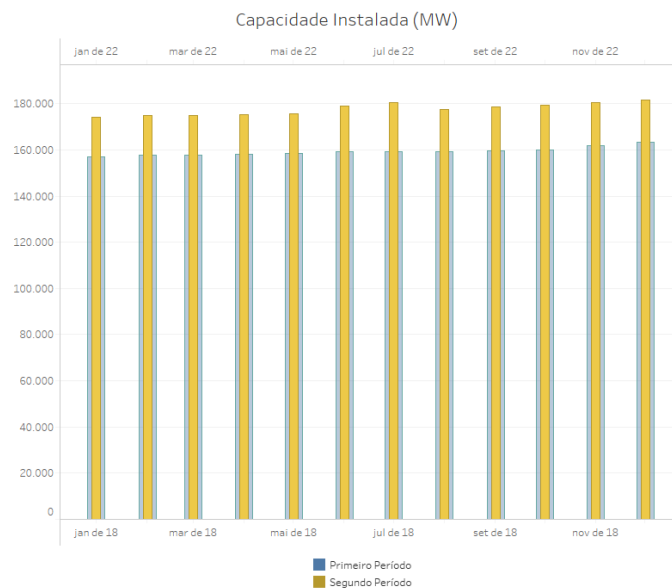
de Energia Solar teve um aumento significativo ao se comparar com a Energia Hídrica, isso pode ser explicado pelo fato do custo da tecnologia fotovoltaica ter sofrido uma queda de 20% em 2023 em relação ao mesmo período do ano anterior, tornando-a mais acessível [6], bem como pelo fato da exploração hidráulica no país, na atualidade, gerar impactos ambientais de grande monta. A Figura 2 mostra a capacidade instalada de Energia solar ao decorrer dos anos e a Figura 3 uma comparação.

Figura 2 - Capacidade Instalada de Energia Solar entre abril de 2018 e março de 2023



Fonte: [2]

Figura 3 - Comparação da Capacidade Instalada de Energia Hídrica de 2018 e 2022



Fonte: [2]

As fontes renováveis causam danos menores à atmosfera quando comparadas com a energia advinda da queima de combustíveis fósseis e são as principais aliadas para contribuir no processo de descarbonização. Considerando que há uma busca pela redução da quantidade de gases emitida para a atmosfera a médio e longo prazos, tem-se que a energia solar é uma boa proposta de investimento para esse processo. Neste ínterim, sabe-se que a principal causa do crescimento do efeito estufa está relacionada com o aumento nos níveis de emissão de gás carbônico na atmosfera terrestre [7]. Na Figura 4 é apresentado benefícios em relação a descarbonização.

Figura 4 - Benefícios da Descarbonização



Fonte: [7]

Há desafios a serem considerados no processo de descarbonização, pois os sistemas de energia não são compostos apenas por tecnologia e viabilidade econômica, mas também envolve interesses de setores tradicionais da sociedade, logo é preciso ter estratégias, pois essas fontes renováveis estão em conflito com a geração de energia baseada em hidrocarbonetos [8].

Percebe-se que nos últimos anos, houve um aumento na conscientização e ampliação na disseminação do conhecimento sobre o assunto em busca de conter os impactos ambientais e obter mais investimentos na transição para fontes de energias limpas. Assim, energia solar tem recebido muitos incentivos por contribuir com o processo de descarbonização, tais como: subsídios na instalação de sistemas fotovoltaicos e leilões de energia reserva. Com isso o investimento financeiro vem aumentando. Ademais, nota-se outras vantagens na implantação

de sistemas fotovoltaicos, entre elas: o baixo custo de manutenção, vida útil de 25 a 30 anos e por ser uma fonte de energia abundante. Nestes termos, realizando um melhor aproveitamento dos espaços e integrando-a com outras plantas de energia é possível impedir a emissão de milhares de toneladas de gás carbônico na atmosfera [8].

1.1 MOTIVAÇÕES

As fontes renováveis são dependentes de condições climáticas, o que pode resultar em intermitência, incerteza e variações diárias ou sazonais. Em determinados momentos, é possível que alguma das fontes de energia esteja menos disponível, enquanto outra esteja com maior disponibilidade de forma que se complementem. Nesse contexto, uma maneira de garantir maior estabilidade na geração de energia elétrica é integrar, através de um sistema híbrido, diferentes tipos de fontes. Neste ínterim, existem alguns arranjos possíveis, entre os quais está a combinação entre hidroeletricidade e energia solar fotovoltaica, a qual será foco deste trabalho, tendo em vista o alto número de usinas hidrelétricas instaladas no Brasil e o potencial da energia solar no país.

Para realizar a análise de complementaridade de fontes em uma determinada região e verificar a viabilidade do sistema híbrido é necessário avaliar a geração de energia proveniente das duas distintas fontes. No caso da geração hidráulica, o planejamento da operação feita pelo ONS está, na atualidade, embasado na utilização dos programas do CEPTEL (Centro de Pesquisas de Energia Elétrica): Newave, Decomp e Dessem. Enquanto para a fotovoltaica percebe-se, na atualidade, uma carência de ferramentas que contemplem algoritmos próprios de previsibilidade para a estimação de irradiância e, conseqüentemente, estimação de geração de energia futura. Dessa forma, o estudo busca ampliar essa análise com investigação da previsibilidade futura de geração solar, realizando uma comparação entre modelos estatísticos já utilizados e algoritmos de previsibilidade.

1.2 OBJETIVO GERAL

O trabalho tem como objetivo contextualizar e descrever os tipos de usinas híbridas, com foco na hidráulica e fotovoltaica e os empecilhos deste tipo de projeto, apontando regulação, vantagens, desvantagens e desafios. Pretende-se, também, analisar, selecionar e implementar diferentes algoritmos para a previsibilidade de geração solar a curto e médio prazo no contexto de hibridização de fontes, utilizando a linguagem Python e verificar quais

algoritmos se apresentam com maior eficiência ao comparar com os modelos estatísticos usados em análise de séries temporais já implementados na literatura. Por fim, busca-se realizar análise crítica e propositiva acerca do uso de algoritmos sofisticados na previsibilidade de geração fotovoltaica.

1.3 ESTRUTURA DO TRABALHO

Com o objetivo de atingir as metas estabelecidas no trabalho, foram estruturados capítulos que apresentam os temas de forma a favorecer a evolução das ideias. Assim sendo, a monografia será organizada da seguinte forma:

O Capítulo 1 introduz uma análise da matriz energética do Brasil, com enfoque nas fontes renováveis, especialmente as energias solar e hídrica, destacando sua contribuição no processo de descarbonização.

No Capítulo 2, são apresentados conceitos fundamentais acerca do sistema híbrido, abordando sua regulação, desafios, bem como a combinação entre hidroeletricidade e fotovoltaica. É apresentado um exemplo já implementado para melhor compreensão da relação entre a demanda e a geração acumulada no sistema. Também são apresentados os potenciais de energia hidráulica e solar no país. Além disso, o capítulo apresenta modelos estatísticos utilizados em análise de séries temporais, já implementados no sistema, assim como algoritmos de previsibilidade existentes na literatura científica correspondente.

O Capítulo 3 apresenta a implementação computacional dos algoritmos de previsibilidade selecionados utilizando a linguagem Python, a partir dos dados disponíveis no Instituto Nacional de Meteorologia (INMET). Adicionalmente, são implementados modelos estáticos tradicionais que já são utilizados atualmente. O último tópico desse capítulo busca validar a eficácia dos algoritmos e para isso são realizadas comparações entre os dados obtidos no modelo estático e os algoritmos de previsibilidade. Com base nesses resultados, é realizada uma análise crítica sobre a utilização desses algoritmos de previsibilidade de geração solar, visando seu emprego na análise de complementaridade de fontes no contexto da implementação de sistemas híbridos de energia renovável.

Por fim, o Capítulo 5 apresenta as conclusões referentes ao estudo realizado, abrangendo as comparações feitas e os resultados obtidos. Com base nesses resultados, é realizada uma análise crítica sobre a utilização desses algoritmos de previsibilidade de geração solar, visando seu emprego na análise de complementaridade de fontes no contexto

da implementação de sistemas híbridos de energia renovável. Além disso, são apresentadas sugestões para trabalhos futuros e pesquisas futuras na área.

2 FUNDAMENTAÇÃO TEÓRICA

O capítulo em questão tem como objetivo apresentar a fundamentação teórica que é necessária para auxiliar no desenvolvimento e compreensão do tema abordado. Para isso, foram selecionados trabalhos publicados nos últimos anos que serviram como referência. Dentre essas referências, foram selecionados artigos que abordam temas como sistema híbridos, potencial de energia hidráulica e solar no país, modelos estatísticos e algoritmos de previsibilidade.

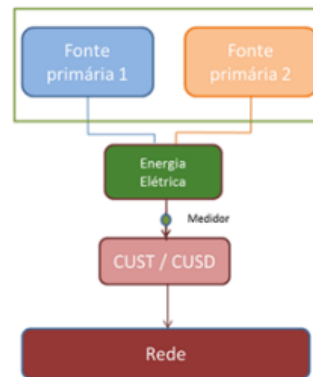
2.1 SISTEMA HÍBRIDO

De acordo com a Agência Nacional de Energia Elétrica (ANEEL), uma usina híbrida pode ser definida como “um sistema que combina duas ou mais formas de produção de energia ou de potência”. Logo, ela possui uma única autorização, podendo ter medições distintas por tecnologia ou únicas, sendo assim, ela tem o uso da rede de transmissão compartilhada entre as fontes [9].

Com isso, ao buscar um melhor aproveitamento do sistema de transmissão já existente, realizando a combinação entre usinas novas e existentes, as usinas que utilizam mais de uma fonte primária ganharam destaque, pois a eventual complementariedade natural existente entre elas pode tornar possível ampliar a disponibilidade energética na matriz energética do Brasil. Além disso, existe um aproveitamento do mesmo espaço físico e o compartilhamento de equipamentos do sistema e de geração, assim como O&M, diminuindo assim uma parcela do custo total do projeto [10].

É importante compreender a distinção entre usinas híbridas e usinas associadas. Uma usina geradora híbrida é responsável por instalar sistemas de geração de energia elétrica que combinam diferentes tecnologias, podendo ter medições separadas por tecnologia ou não. Esse tipo de usina é representado na Figura 5. Por outro lado, as usinas geradoras associadas utilizam duas ou mais instalações para produzir energia elétrica com tecnologias de geração diferentes. Cada instalação tem outorgas e medições distintas, mas compartilham fisicamente e contratualmente a infraestrutura de conexão e uso do sistema de transmissão. Esse tipo de usina é representado na Figura 6 [4].

Figura 5 - Usina Híbrida



Fonte: [10]

Figura 6 - Usina Associada



Fonte: [10]

2.1.1 REGULAÇÃO

As regulações e normas, no tocante à sistema híbrido ou associados, foram discutidas pela ANEEL no período de 2018 a 2019, porém antes disso ocorreram alguns projetos com combinação de geração fotovoltaica, hidroeletricidade e parques eólicos [9].

Para que seja permitida a associação de uma usina a outra já existente, é necessário que elas possuam outorgas, medições e monitoramento distintos para cada fonte, pois as usinas já existentes possuem contratos prévios de uso firmados, então é preciso ter uma boa estruturação do projeto [11]. Além disso, é importante realizar estudos técnicos para avaliar se tem possibilidade de ocorrer a integração de duas ou mais usinas, assim como verificar a viabilidade econômica.

Para o desenvolvimento de projetos híbridos, é necessário estabelecer critérios, como potência instalada ou a geração de energia futura. No caso da geração hidráulica, já existem

programas do Centro de Pesquisas de Energia Elétrica (CEPEL) com essa finalidade, porém para a fotovoltaica há uma carência de ferramentas capazes de estimar essa geração. Com isso, o trabalho em questão busca avaliar o desempenho de modelos estatísticos e algoritmos de previsibilidade, capazes de ajudar na regulação desse tipo de projeto, viabilizando assim o desenvolvimento de novas usinas híbridas.

2.1.2 DESAFIOS

Existem alguns desafios em um processo de hibridização, caso os recursos de energia disponíveis excederem a capacidade de energia híbrida, existirá o risco de usar poucos recursos. Além disso, ao ocorrer uma falha técnica em uma usina híbrida pode vir a afetar a produção de diversas fontes ao mesmo tempo. Além do mais, outro ponto crítico é a necessidade de se estabelecer um projeto de modo que uma fonte de energia não interfira na outra [12].

2.1.3 COMBINAÇÃO ENTRE HIDROELÉTRICA E FOTOVOLTAICA

Em determinados momentos é possível que alguma das fontes de energia esteja menos disponível, enquanto outra esteja com maior disponibilidade de forma que se complementem. Logo, para ter uma geração de energia elétrica mais estável é possível fazer uma integração através de um sistema híbrido.

Existem algumas combinações possíveis de fontes energéticas que compõe usinas híbridas, sendo algumas delas listadas pela Empresa de Pesquisa Energética (EPE) no documento de Estudos de Planejamento da Expansão da Geração [10]. Podendo-se citar: a junção da eólica com energia solar, heliotérmica com biomassa, gás natural com biomassa, carvão com biomassa e hidroeletricidade com fotovoltaica que será aprofundada nesse trabalho.

A combinação entre usina hidrelétrica com energia solar pode ser feita através de painéis fotovoltaicos flutuadores, dessa forma, é possível obter um aproveitamento mais eficiente na região inundada que, em geral, não são utilizados. Além disso, há estudos que demonstram que essa associação pode ter um aumento de 17% da capacidade de geração de energia elétrica do arranjo [13], inclusive pelo fato de reduzir a evaporação no reservatório e elevar a quantidade de água armazenada para geração de energia futura. O grande potencial desse tipo de arranjo está no fato de que usinas hidrelétricas podem ter sua geração ajustada

atuando como compensador em tempo real para a intermitência da geração fotovoltaica, na Figura 7 um exemplo de parque solar flutuante.

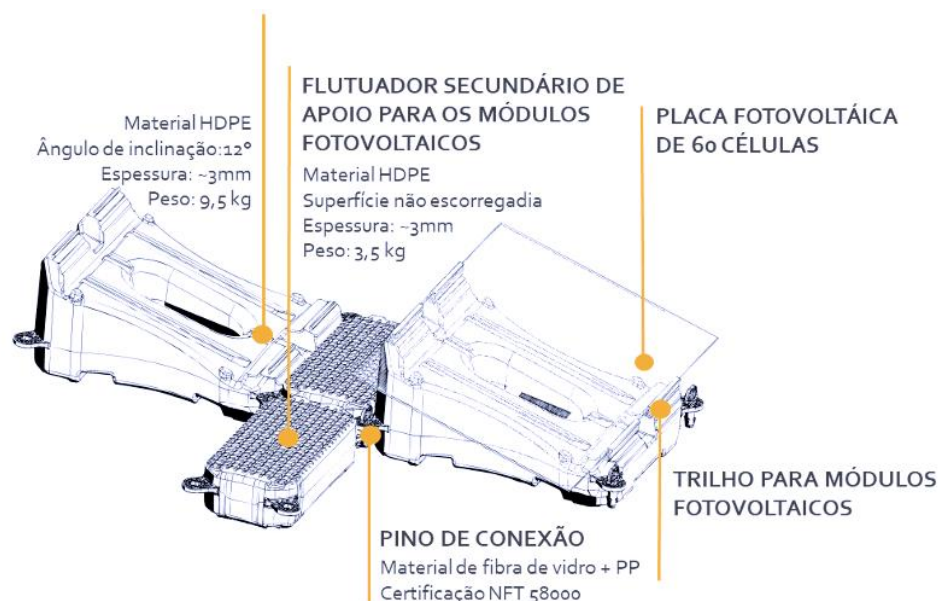
Figura 7 - Energia solar fotovoltaica flutuante em Rosana, São Paulo



Fonte: [14]

Para realizar essa combinação é essencial utilizar estruturas que sustentam os painéis solares acima da água, essas estruturas são chamadas de flutuadores. Atualmente, há flutuadores com suporte de metal que são fixados ao módulo e os flutuadores puros desenvolvido para receber a placa diretamente, com isso o plástico é reforçado com fibra de vidro, o esquema de montagem está representado na Figura 8.

Figura 8 - Esquema de montagem dos painéis solares sobre a água



Fonte: [15]

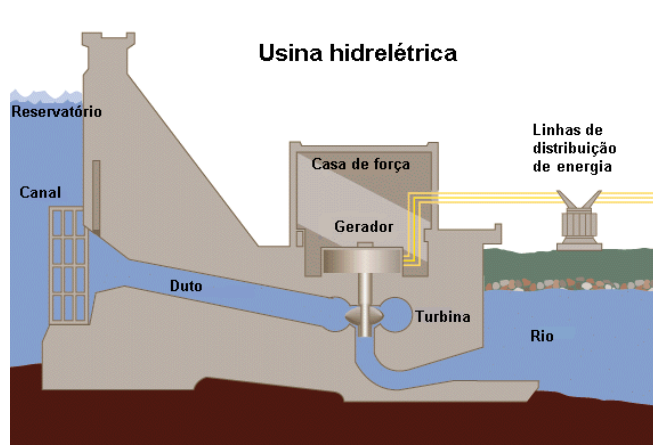
2.2 POTENCIAL DE ENERGIA HIDRÁULICA NO PAÍS

De acordo com dados da Agência Nacional de Energia Elétrica (ANEEL), o Brasil possui um enorme potencial de energia hidráulica, graças as diversas bacias hidrográficas e sua ampla rede fluvial. Com isso, ela é a principal fonte utilizada no país, sendo responsável por cerca de 60% da capacidade de geração de energia elétrica.

É importante destacar que mesmo sendo uma fonte renovável e limpa, ela pode ter impactos relevantes no meio ambiente e nas comunidades locais, então é preciso ter um equilíbrio entre a preservação do meio ambiente e comunidade com a geração de energia hidrelétrica. Caso haja um planejamento apropriado é possível diminuir os efeitos negativos causados por ela [16].

Para ocorrer o funcionamento de uma usina hidrelétrica de armazenamento é preciso ter um reservatório composto por uma barragem ou represa. Quando a água é liberada do reservatório, ela flui em direção as turbinas e com a força da água movimentada as pás dela. Isso faz com que os geradores sejam ativados, convertendo, assim, energia potencial gravitacional em eletricidade [16]. A Figura 9 é um exemplo de usina hidrelétrica de armazenamento.

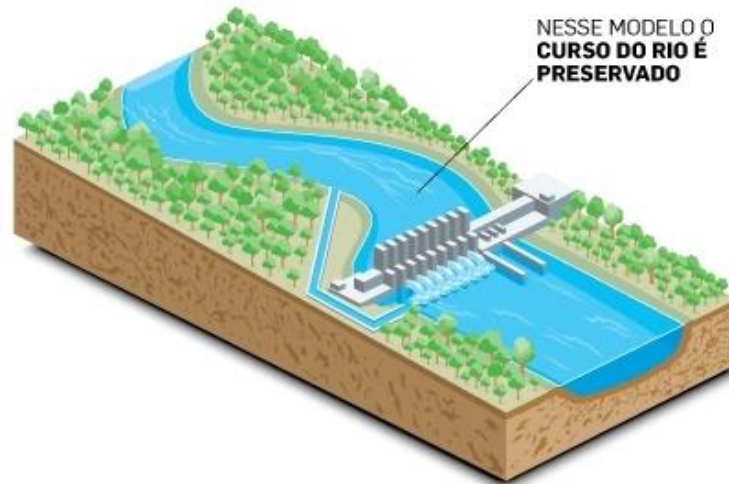
Figura 9 - Usina hidrelétrica de armazenamento



Fonte: [17]

No caso da usina hidrelétrica a fio d'água, não há necessidade de estocagem de água, pois utiliza, em geral, a força da correnteza dos rios de grande altitude e baixa vazão. No entanto, durante períodos com pouca chuva, esse tipo de usina pode ficar sem água, diminuindo a autossuficiência energética [18]. A Figura 10 é um exemplo de Usina hidrelétrica a fio d'água.

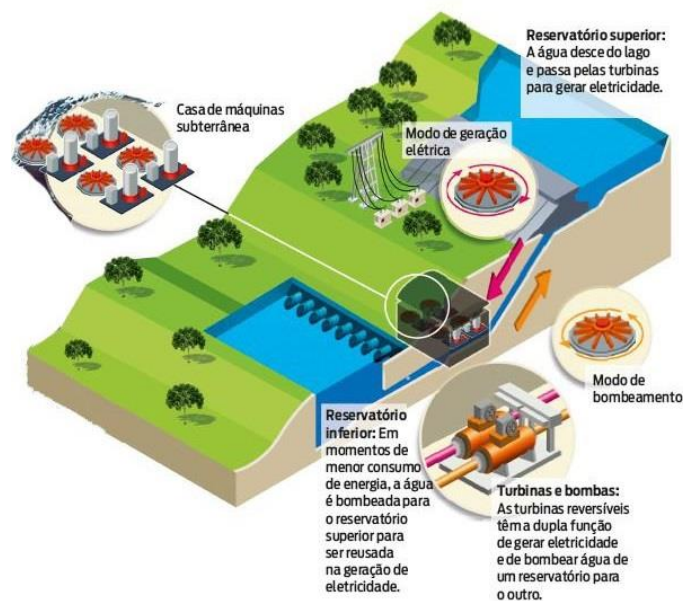
Figura 10 - Usina hidrelétrica a fio d'água



Fonte: [19]

Além das usinas mencionadas, existe também a opção da usina hidrelétrica reversível. Seu funcionamento pode ocorrer de forma que a usina seja geradora ou bombeadora. Ela é composta por um reservatório inferior e um superior, onde através de um dispositivo de elevação de água, a água do inferior é bombeada para o superior. Quando ocorre a queda da água armazenada as turbinas são ligadas, produzindo eletricidade [20]. A Figura 11 é um exemplo de usina hidrelétrica reversível.

Figura 11 - Usina hidrelétrica reversível



Fonte: [21]

Independentemente do tipo de usina hidrelétrica, ao se basear na equação de Bernoulli, é deduzida a fórmula da potência teórica entre dois pontos de um curso de água, e ao multiplicar pelo rendimento total da usina, se tem a potência de uma usina hidrelétrica em condições ideais sem considerar nenhuma perda, sendo a Equação 1:

$$P = \rho \cdot g \cdot Q \cdot \Delta h \cdot \eta \quad (1)$$

Sendo que:

P = Potência de uma usina hidrelétrica [W];

ρ = Densidade da água [Kg/m³];

g = Aceleração da gravidade [9,81 m/s²];

Q = Vazão da água [m³/s];

Δh = Altura da queda da água [m];

η = Rendimento total.

O planejamento da operação do sistema elétrico brasileiro é realizado pelo Operador Nacional do Sistema Elétrico (ONS). É possível definir a geração de energia elétrica de cada usina hidrelétrica através de simulações computacionais, modelos matemáticos e as previsões de demanda por energia elétrica. Além disso, é realizado o acompanhamento em tempo real da geração de energia.

Utilizando os programas do Centro de Pesquisas de Energia Elétrica (CEPEL): Newave, Decomp e Dessem, é calculado pela Câmara de Comercialização de Energia Elétrica (CCEE) o Preço de Liquidação das Diferenças (PLD), o PLD determina a diferença entre a energia elétrica que não foi contratada pelos negociantes e o valor total da energia gerada, e esse valor tem como base o Custo Marginal de Operação (CMO).

Com isso, o cálculo tem como objetivo priorizar armazenamentos futuros da água e a economia que isso irá gerar, pois terá uma redução na queima de combustíveis em usinas termoelétricas. Consequentemente, os cálculos do PLD são baseados em previsões e suposições. As suposições incluem informações presumidas, ou seja, antes da operação real do sistema. São considerados a disponibilidade declarada de geração de eletricidade e os valores de consumo previstos para cada nicho de mercado [22]. Na Figura 12 tem o fluxograma dos prazos dos modelos computacionais.

Figura 12 - Prazos dos modelos computacionais.



Fonte: [22]

2.3 POTENCIAL DE ENERGIA SOLAR NO PAÍS

O Brasil apresenta as condições ideais para estar entre os cinco países que mais fornecem energia solar do mundo. Mesmo que em algumas partes a incidência solar seja menor, o país possui um vasto território de dimensões continentais, com isso, a maior parte das regiões apresentam mais de 10 horas diária de sol, evidenciando seu alto potencial [23]. Trata-se de um país tropical. O Noroeste de Minas Gerais e a região central da Bahia possuem os mais altos valores de irradiação solar, chegando a 6,5 KWh/m² por dia [24].

Para ocorrer o funcionamento de uma usina solar é necessário ter instalado os módulos fotovoltaicos, que convertem a luz em energia elétrica através do efeito fotovoltaico. O efeito fotovoltaico ocorre quando as células fotovoltaicas compostas por materiais semicondutores presentes na placa solar são submetidas as partículas de luz da radiação, chamadas fótons, eles excitam os elétrons dos átomos, provocando seu movimento através das diversas camadas do dispositivo e gerando uma corrente elétrica contínua. Após isso, o Inversor transforma corrente contínua em corrente alternada [25].

A potência de uma usina Solar, pode ser calculada utilizando a Equação 2:

$$P = \eta \cdot A \cdot R_s \cdot N \quad (2)$$

Sendo que:

P = Potência de uma usina solar [W];

η = Rendimento total;

A = Área do painel [m²];

R_s = Radiação solar incidente sobre os painéis [W/m²];

N = Quantidade de painéis.

É importante destacar que para obter cálculos mais precisos, é aconselhável realizar a medição da radiação solar na região em que o sistema fotovoltaico será instalado, utilizando o Piranômetro, porém é preciso levar em consideração as diferentes épocas do ano e as horas ao decorrer do dia. Com isso, para facilitar é possível pegar dados em sites meteorológicos já disponibilizados [24].

Para prever a geração solar futura em diferentes condições climáticas o ONS utiliza dados históricos meteorológicos de radiação solar e temperatura do local, para assegurar uma estimativa de geração solar [26]. Com os modelos estatísticos é capturado tendências e padrões a longo prazo, porém esses são os modelos tradicionais, existem algoritmos de previsibilidade mais atuais que podem vir a ser implementados.

2.4 MODELO DE PREVISÃO ESTATÍSTICO

Os modelos de previsão estatísticos utilizam uma série de dados históricos para efetuar prognósticos fundamentados, a partir dos dados passados coletados [27]. Esses modelos são implementados, pois proporcionam flexibilidade para se ajustarem às mudanças nos padrões de tendência e sazonalidade [28].

Atualmente, para se ter uma previsão da geração solar são utilizados esses métodos de previsão, nos quais são utilizados dados meteorológicos de anos anteriores, tornando possível ter uma estimativa da geração futura.

Nos próximos dois subtópicos, serão abordados modelos de previsão estatístico tradicionais, que são alguns dos métodos utilizados por Engenheiros e Pesquisadores. Com o propósito de evidenciar a existência de diferentes formas de escrever as equações que descrevem o comportamento do sistema [29].

2.4.1 ARMA

O modelo Autoregressive Moving Average (ARMA) pode ser traduzido como Média Móvel Autorregressiva, e é a combinação de um Modelo Autorregressivo (AR) com o modelo de Dados de Médias Móveis (MA). É o modelo mais conhecido para séries temporais univariadas com valores de 'p' e 'q', e foi criado em 1970 por George Box e Gwilym Jenkins [30].

Esses dois modelos partem do pressuposto de que a sequência temporal é gerada por meio de um sistema linear e são um sinal aleatório conhecido como ruído branco, possuindo média e variância constante, com isso, não apresentam um padrão [31].

Porém, o modelo ARMA tem uma significativa simplificação na previsão linear, uma vez que possui a capacidade de modelar um processo não conhecido, com o mínimo de parâmetros necessários [32].

O Modelo Autorregressivo de ordem p ($AR_{(p)}$) é definido pela Equação 3 a seguir [33].

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t \quad (3)$$

Sendo que:

Z_t = Combinação linear de valores passados com o termo a_t , ou seja, corresponde ao valor da série temporal no instante t;

Z_{t-p} = São termos independentes que indicam a relação entre Z_t e valores passados;

a_t = Leva em conta fatores presentes na série até o tempo t, que não são explicados pelos valores Z_{t-p} (passados).

Ao considerar a série Z_t e fazendo os $\Psi^0 s = -\theta^0 s$ na abordagem do processo linear geral, assume-se que a série temporal é finita e a equação é representada de forma linear. Esse modelo é conhecido como Médias Móveis de ordem q ($MA_{(q)}$), e é definido pela Equação 4 a seguir [33].

$$Z_t = a_t - \theta_{1at-1} - \theta_{2at-2} - \dots - \theta_{qat-q} a_t \quad (4)$$

Sendo que:

Z_t = Combinação linear dos pesos estabelecidos com o termo a_t , ou seja, corresponde ao valor da série temporal no instante t;

θ_{qat-q} = Pesos estabelecidos aos erros em cada instante de tempo passado;

a_t = Erro no tempo t, sendo ele um valor aleatório.

Com as duas equações anteriores definidas, é possível definir a Equação 5 $ARMA_{(p,q)}$:

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t - \theta_{1a} a_{t-1} - \theta_{2a} a_{t-2} - \dots - \theta_{qa} a_{t-q} \quad (5)$$

2.4.2 ARIMA

O modelo Autoregressive Integrated Moving Average (ARIMA), criado em 1970 por Box e Jenkins [31], pode ser traduzido como Média Móvel Integrada Autorregressiva, e além da combinação de um Modelo Autorregressivo (AR) com o modelo de Dados de Médias Móveis (MA), esse modelo também utiliza a Diferenciação (I).

Esse modelo é utilizado por empresas de previsão de demanda, bancos, por agências de planejamento energético, institutos de pesquisa e departamentos de estatística.

Esse modelo também é univariado, com valores de 'p', 'd' e 'q', e para determinar esses valores é verificado se há necessidade de transformação de uma série exponencial para uma linear e para a tornar a série estacionária é preciso diferenciá-la 'd' vezes [34].

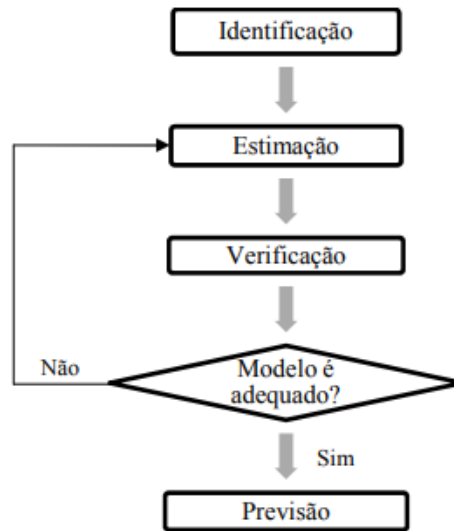
Para utilizar o modelo autorregressivo de médias móveis, se faz necessário ajustá-lo ao conjunto de dados históricos. Com isso, é realizado um ciclo iterativo como forma de pré-processamento.

É realizada uma especificação, onde alguns modelos são considerados para a análise. Feito isso, é identificado o modelo de autocorrelação que descreve o comportamento da série. Após isso, se estima os parâmetros, onde as estimativas iniciais obtidas durante a fase de identificação serão utilizadas como valores iniciais nesse processo. Por último, é feita a verificação por meio de uma análise de resíduos [34].

Com todos os passos anteriores realizados é possível realizar a previsão. Logo, com a base da equação $ARMA_{(p,q)}$, é possível definir a Equação 6 $ARIMA_{(p,d,q)}$:

$$\Delta^d Z_t = W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \dots + \phi_p W_{t-p} + a_t - \theta_{1a} a_{t-1} - \theta_{2a} a_{t-2} - \dots - \theta_{qa} a_{t-q} \quad (6)$$

Figura 13 - Ciclo iterativo do modelo ARIMA



Fonte: [31]

2.4.3 ARIMAX

O modelo Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) pode ser traduzido como Média Móvel Integrada Autorregressiva com variável exógena, é uma extensão do modelo ARIMA citado no subtópico anterior [35].

Esse modelo incorpora uma componente linear que fica em função das observações das demais variáveis, sendo ela chamada de Variável Exógena ($X_{(r)}$). Com isso, o modelo $ARIMAX_{(p,d,q,r)}$, pode ser definido pela Equação 7:

$$y_t = \rho + \sum_{i=1}^p \beta_i y_{t-i} + \sum_{j=1}^r \omega_j w_j + \sum_{j=1}^q (\theta_j - \varepsilon_{t-j}) + \varepsilon_t \quad (7)$$

Sendo que:

y_t = Variável dependente no tempo t ;

ρ = Constante;

y_{t-i} = Variável com atraso no tempo;

β_i = Coeficiente da variável com atraso no tempo;

p = Número máximo de períodos;

w_j = Variável exógena;

ω_j = Coeficiente da variável exógena;

r = Número máximo de variáveis exógenas;

ε_t = Erro no tempo t , podendo ser defasado em j ;

θ_j = Coeficiente do erro.

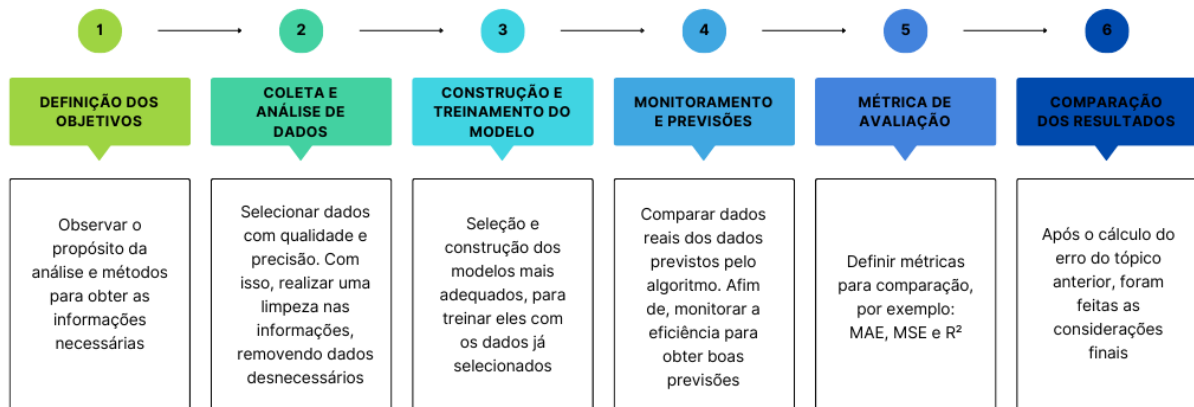
Esse modelo é utilizado atualmente pelo Operador Nacional do Sistema Elétrico (ONS), para prever a geração solar. Através dele são realizadas análises dos indicadores de previsão e a previsão de geração fotovoltaica para a programação diária. Além do Python, é utilizada a linguagem de programação R.

2.5 ALGORITMOS PARA PREVISIBILIDADE

Uma ferramenta importante são os algoritmos de previsibilidade, pois permitem mapear comportamentos e podem ser aplicados em diversas áreas. Com isso, através de estimativas e dados históricos, é possível realizar prognósticos futuros.

Para entender como funciona e como se cria um algoritmo de previsibilidade, é importante seguir algumas etapas, sendo elas mostradas na Figura 14 [36].

Figura 14 - Como criar um algoritmo de previsibilidade.



Fonte: Própria Autora

Abaixo são mencionados alguns exemplos de algoritmos comuns:

- Algoritmo de pesquisa: Pode ser uma pesquisa sequencial ou de intervalo, onde busca dados específicos entre os dados armazenados.
- Algoritmo de ordenação: Organiza os dados em uma ordem específica, podendo ser crescente ou decrescente.

- Algoritmo de Compressão: Utilizados para reduzir tamanho de dados ou arquivos.
- Algoritmo de Criptografia: Utilizado para criptografar assinaturas e mensagens, onde os algoritmos definem o Uniform Resource Identifier (URI).
- Algoritmo de aprendizado de máquina: Tem como objetivo realizar previsões utilizando série de dados.

Com base nos exemplos mencionados, o algoritmo de aprendizado de máquina é uma excelente ferramenta para realizar análise com estudos de previsibilidade futura. No setor de energia tem sido observado que esse recurso tem se mostrado muito eficaz, isso tem permitido assegurar a segurança energética e fornece parâmetros valiosos para orientar decisões importantes.

O aprendizado de máquina é uma parte da inteligência artificial, sendo responsável por aprender padrões complexos, e com isso executar atividades a partir dos dados acumulados de soluções passadas [37].

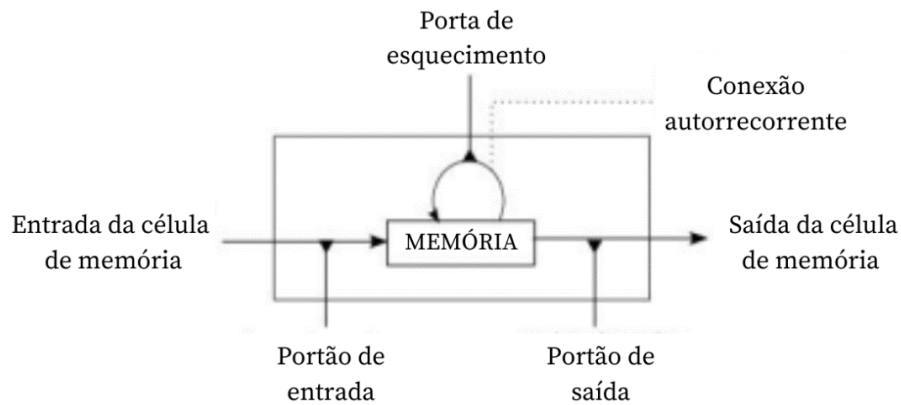
Existem diversas técnicas de aprendizado de máquina disponíveis na literatura científica, e a seleção dessas técnicas deve ser feita com base na natureza específica do problema que se deseja resolver [38]. A partir disso, nos subtópicos a seguir foram apontados dois exemplos desses algoritmos de previsibilidade, onde ambos são modelos de redes neurais que utilizam a ideia de atraso no tempo e outro algoritmo de rede neural que não utiliza dessa ideia.

2.6.1 LSTM

O algoritmo Long Short Term Memory (LSTM) pode ser traduzido como Memória de Longo Prazo. Ele é um dos algoritmos de redes neurais mais avançados e utilizados, pois apresenta uma expressiva diminuição nos erros de previsão ao lidar com dados temporais que apresentam variações intensas, devido a mudança de temperatura, a interferência das nuvens afetando a radiação solar, entre outros aspectos [39].

Esse método utilizado em redes neurais é composto por uma célula de memória e unidades de porta. Essa célula de memória possui uma memória explícita, onde mantém e atualiza os dados históricos ao decorrer do tempo, e as unidades de porta controlam o fluxo de informação para dentro e fora da memória [40]. A Figura 15 mostra uma célula de memória.

Figura 15 - Célula de memória.



Fonte: [40]

A célula de memória utiliza portões com o objetivo de reduzir o peso de entrada, controlando o erro. Com isso, os portões podem escolher quando mantém ou substituem as informações.

A porta de esquecimento (*forget gate*) é responsável por controlar as informações que serão descartadas da célula de memória [40]. E é expressa pela Equação 8:

$$f_t = \sigma.(W_f.[h_{t-1}, x_t] + b_f) \quad (8)$$

Sendo:

f_t = Porta de esquecimento;

σ = Função logística;

W_f = Peso da porta de esquecimento;

h_{t-1} = Estado oculto no tempo anterior;

x_t = Entrada no tempo atual;

b_f = Bias da porta de esquecimento.

O portão de entrada (*Input Gate*) é responsável por regular quais dados são adicionadas ao estado da célula com base na entrada atual [40]. Equação 9:

$$i_t = \sigma.(W_i.[h_{t-1}, x_t] + b_i) \quad (9)$$

Sendo:

i_t = Porta de entrada;

W_i = Peso da porta de entrada;

b_i = Bias da porta de entrada.

Porém como é regulado quais dados são adicionados, a célula de memória realiza uma atualização na memória, onde ela agrega dois elementos à memória anterior por meio da porta de esquecimento, enquanto a nova memória é introduzida na porta de entrada [40]. Sendo essa atualização representada pelas Equações 10 e 11:

$$\hat{C}_t = \tanh.(W_C . [h_{t-1}, x_t] + b_C) \quad (10)$$

e

$$C_t = f_t . C_{t-1} + i_t . \hat{C}_t \quad (11)$$

Sendo:

\tanh = Função tangente hiperbólica;

W_C = Peso da memória de atualização;

b_C = Bias da atualização;

C_{t-1} = Memória no tempo anterior;

O Portão de saída (*Output Gate*) é responsável por decidir o que será produzido de memória [40]. Sendo a Equação 12:

$$o_t = \sigma.(W_o . [h_{t-1}, x_t] + b_o) \quad (12)$$

Em que:

o_t = Porta de saída;

W_o = Peso da porta de saída;

b_o = Bias da porta de saída.

Com isso, se tem a saída da célula de memória [40]. Representada pela Equação 13 final:

$$h_t = o_t \cdot \tanh(C_t) \quad (13)$$

Sendo:

h_t = Saída da célula de memória.

2.6.2 GRU

O algoritmo *Gated Recurrent Unit* (GRU) pode ser traduzido como Rede Neural de Unidade Recorrente Fechada. É uma variação do algoritmo anterior LSTM e ambas geram resultados parecidos, a Rede Neural Recorrente foi criada em 2014 por Kyunghyun Cho, enquanto a LSTM foi criada em 1997. A criação do algoritmo teve como foco melhorar os resultados, diminuindo a quantidade de portas, com isso, deixando-o mais compreensível [37].

Esse método foi criado para utilizar memória de curto prazo, onde existe somente uma única camada oculta nele, responsável por armazenar a memória de curto e longo prazo. E é estruturado por portões compostos por vetores que variam seu valor entre 0 e 1, que são os responsáveis por controlar todas as informações, incluindo as que são excluídas e as que são mantidas, ou seja, quando se tem 1 quer dizer que o valor é importante e não será descartado. A Rede Neural GRU utiliza o portão reiniciar (*reset gate*) e O portão de atualização (*update gate*) para filtrarem quais dados serão passados para a saída. Além disso, é importante saber que quando se tem uma grande sequência de dados o algoritmo consegue capturar informações de etapas anteriores sem que elas sejam descartadas [41].

O portão reiniciar (*reset gate*) é responsável por controlar a influência do estado oculto no tempo anterior [41]. E é expresso pela Equação 14:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (14)$$

Sendo:

r_t = Portão reiniciar;

W_r = Parâmetro de peso, aprendido durante o treinamento;

h_{t-1} = Estado oculto no tempo anterior;

x_t = Entrada no tempo atual.

A portão de atualização (*update gate*) é responsável por controlar quais informações serão ignoradas [41]. E é expressa pela Equação 15:

$$u_t = \sigma(W_u \cdot [h_{t-1}, x_t]) \quad (15)$$

Sendo:

u_t = Portão de atualização;

W_u = Parâmetro de peso, aprendido durante o treinamento;

Logo, a equação de saída é dada pelas Equações 16 e 17 [41]:

$$\tilde{h} = \tanh.(W \cdot [r_t * h_{t-1}, x_t]) \quad (16)$$

e

$$h_t = (1 - u_t) \cdot h_{t-1} + u_t \cdot \tilde{h}_t \quad (17)$$

Sendo:

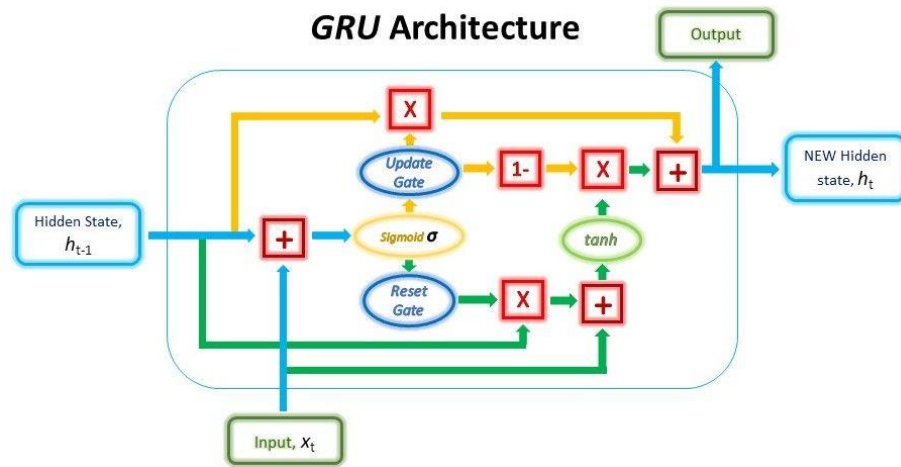
W = Parâmetro de peso;

\tilde{h} = Estado oculto;

h_t = Valor de saída.

A Figura 16 apresenta uma Arquitetura de uma Rede Neural GRU.

Figura 16 - Arquitetura de uma Rede Neural GRU.

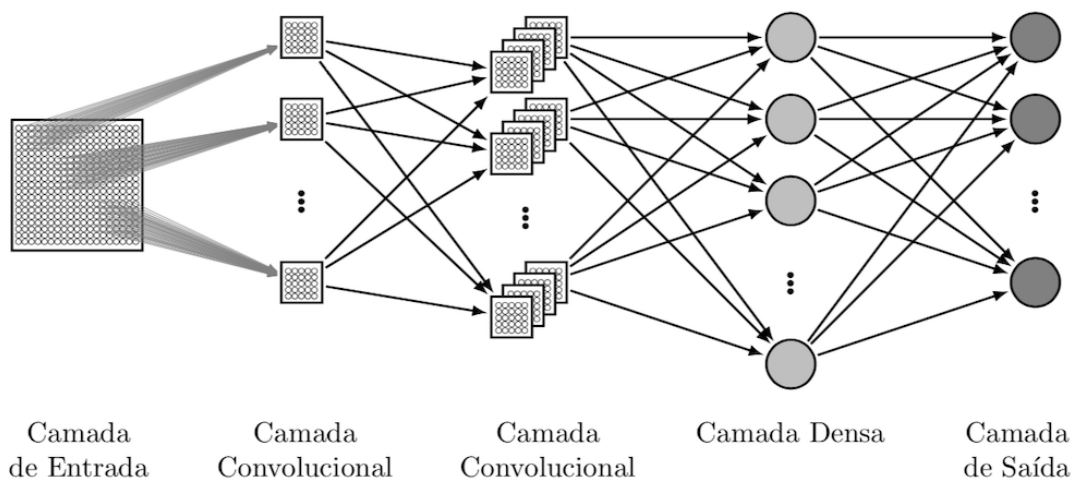


Fonte: [41]

2.6.3 CNN

O algoritmo *Convolutional Neural Network* (CNN) pode ser traduzido como Rede Neural Convolutiva. É um algoritmo muito utilizado em imagens, mas também é implementado em outros casos, os neurônios dele examinam a camada anterior em grupos pequenos e geram uma saída para cada conjunto examinado. Logo, ele é capaz de identificar padrões complexos presentes dentro dos conjuntos [42]. A Figura 17 mostra as camadas da Rede Neural Convolutiva.

Figura 17 - Camadas da Rede Neural Convolutiva.



Fonte: [43]

Esse algoritmo identifica as características importantes da camada de entrada, por meio de filtros convolucionais e esse processo é realizado no treinamento da rede até que se obter uma característica padrão dos dados. Após isso, é realizado, em outra camada, uma redução desses dados de entrada, o que faz com que não ocupe tanta memória e poupe o processamento da máquina. Então, no final da rede é realizada a saída dos dados [44].

2.7 MÉTRICAS DE AVALIAÇÃO

As métricas de avaliação são implementadas para avaliar os desempenhos dos modelos estatísticos de previsibilidade e os algoritmos de aprendizado de máquina, sendo possível comparar métodos, mensurando qual o modelo mais eficaz. Essas métricas são também utilizadas em outros segmentos.

2.7.1 ERRO MÉDIO ABSOLUTO – MAE

O erro médio absoluto calcula a média das diferenças absolutas entre os valores de teste e os valores de treino nas amostras [45]. A Equação 18 é dada pela seguinte fórmula:

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n} \quad (18)$$

Sendo:

\hat{y}_i = Conjunto de dados do treino;

y_i = Valores de teste previstos pelo modelo;

n = Número de amostras.

2.7.2 COEFICIENTE DE DETERMINAÇÃO – R²

O coeficiente de determinação é uma medida amplamente utilizada para avaliar a eficácia de modelos de previsão, permitindo quantificar a dispersão dos dados em relação à linha de regressão. Essa métrica auxilia a determinar a qualidade do ajuste do modelo aos dados. O coeficiente de determinação varia entre 0 e 1, sendo que valores mais próximos de 1

indicam um melhor ajuste, indicando que o modelo é capaz de explicar a variabilidade presente nos dados. Por outro lado, quando o coeficiente é próximo de zero, isso indica que o modelo não conseguiu explicar adequadamente a variação observada [46]. Sua Equação 19 é representada da seguinte forma:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (19)$$

Sendo:

\hat{y}_i = Conjunto de dados do treino;

y_i = Valores previstos pelo modelo;

\bar{y}_i = Valor médio dos elementos do conjunto de treino.

3 DESENVOLVIMENTO DOS ESTUDOS DE DESEMPENHO

A realização da previsão de geração solar é essencial para o planejamento adequado, pois leva em consideração a oferta e demanda de energia elétrica. E ao falar de um sistema híbrido, é importante potencializar as fontes renováveis para se ter um sistema mais sustentável, ocorrendo uma harmonia entre diferentes fontes.

Para prever a geração solar no contexto de hibridização de fontes para análise de complementaridade é necessário utilizar uma base de dados históricos que contenham informação da radiação solar do local em que a usina será implementada, no caso em questão os dados estão sendo coletados perto de uma usina hidrelétrica.

Com isso, se faz necessário realizar uma seleção dos dados, em conjunto com análise e tratamento deles. Posteriormente, utilizar os modelos matemáticos e os algoritmos de previsibilidade para estimar a geração futura durante o ano, junto a uma curva de média diária, sendo possível analisar o método mais eficaz através das métricas de avaliação.

Os dados coletados e os estudos feitos estão expostos de tal forma que outros pesquisadores possam se aprofundar no estudo em questão. Logo, o intuito deste capítulo é trazer de forma clara os resultados e técnicas utilizadas.

3.1 SELEÇÃO DOS DADOS

A seleção de dados meteorológicos confiáveis e de alta qualidade é uma etapa fundamental no processo, uma vez que visa garantir a precisão das previsões de geração solar. Nesse contexto, dado que o estudo busca explorar a possível complementaridade entre uma usina fotovoltaica e uma usina hidrelétrica, é essencial coletar dados nas proximidades de uma usina hidrelétrica. Essa abordagem permitirá uma análise mais precisa dos resultados de previsão de geração solar.

Dentre as 22 bacias hidrográficas definidas pelo ONS, foi escolhida a bacia do Rio Jequitinhonha para realização do estudo. Ela é localizada entre os estados de Minas Gerais e Bahia e inclui duas Usinas Hidrelétricas (UHEs): Irapé (Presidente Juscelino Kubitschek) e Itapebi. Os dados utilizados são da estação meteorológica automática do INMET denominada Almenara, situada a cerca de 119 km de Itapebi e 210 km de Irapé. Dentre os diversos dados disponíveis (data hora, precipitação, pressão atmosférica, radiação, temperatura, umidade relativa e informações do vento), foram selecionadas data, hora e radiação.

3.2 ANÁLISE E TRATAMENTO DOS DADOS

Inicialmente, houve a conferência do número de dados de cada arquivo e se não havia nenhum horário duplicado ou faltante. Quanto ao horário, foi corrigido para o fuso-horário de Brasília. Os dados de radiação (kJ/m^2) foram convertidos para irradiância (W/m^2).

No processo de tratamento dos dados, foi verificada a existência de valores fisicamente impossíveis, utilizando os mesmos critérios adotados pela rede SONDA (Sistema de Organização Nacional de Dados Ambientais). Dessa forma, foi calculado o limite físico máximo de irradiância. Para o limite mínimo, foi estabelecido -4 e -2 como limite raro. Caso a quantidade de dados que extrapola os limites fosse maior que 1%, a medição seria desconsiderada.

Quanto aos dados inexistentes (NaN) de irradiância entre às 6h e 19h, foram preenchidos na seguinte ordem de prioridade e de possibilidade [1]:

- 1º: resultado da média entre os valores observados naquele horário no dia anterior e no dia seguinte;
- 2º: resultado da média entre os valores das horas anterior e seguinte;
- 3º: resultado da média mensal dos dados medidos naquele horário.

Excetuando-se o primeiro e o último dia do ano. Em que, caso houvesse uma lacuna, a primeira opção era completá-la com a média dos dados das horas anterior e seguinte. E, caso não fosse possível, com a média mensal para o horário faltante.

No que diz respeito aos outliers, a detecção foi feita por meio do método de Tukey que identifica valores discrepantes baseado na subtração dos quartis inferior e superior. A distância entre eles serve como medida de disseminação dos dados semelhante ao desvio padrão [47]. Após a identificação dos outliers, eles são substituídos pela média entre os valores anterior e posterior da medição respectiva [48].

Após esse tratamento, foi gerada uma planilha com os dados horários e de irradiância modificados para cada ano de medições.

3.3 APLICAÇÃO DOS MODELOS E ALGORITMOS DE PREVISIBILIDADE

Para o Trabalho de Conclusão de Curso, foram selecionados cinco modelos para realizar a previsão de geração solar, em que foram levados em consideração o perfil dos

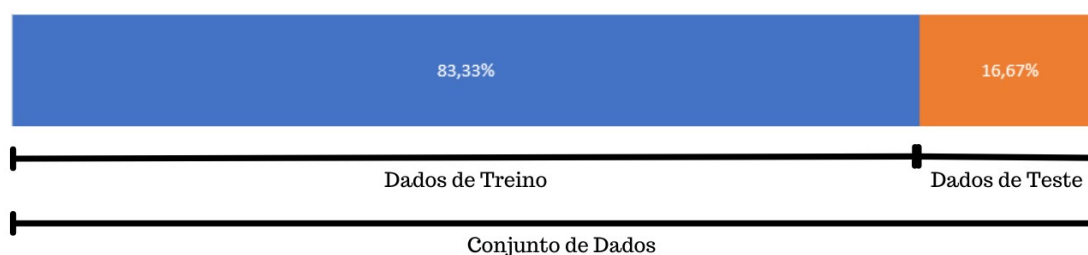
dados, sendo eles: a capacidade em realizar previsões, capacidade em lidar com padrões de dados, como sazonalidade e tendência.

Os modelos foram selecionados com o objetivo de permitir a comparação entre dois modelos estatísticos tradicionais, amplamente empregados, e outros dois algoritmos de redes neurais que são capazes de capturar dependências de longo prazo. Além disso, o último algoritmo, também uma rede neural, possui características distintas dos outros dois, sendo particularmente útil na identificação de padrões nos dados.

Todos os algoritmos foram desenvolvidos utilizando a linguagem Python por meio de um serviço fornecido pelo Google chamado COLABORATORY, conhecido como COLAB, ele permite que o usuário execute e crie códigos pelo navegador, com isso, os arquivos foram armazenados em uma pasta do Google Drive.

Antes de prosseguir com a implementação dos códigos, foi realizada a importação e a instalação das bibliotecas necessárias. Em seguida, os dados foram carregados e plotados. Para o treinamento dos modelos, foram utilizados dados correspondentes aos anos de 2017 a 2021, enquanto os dados de 2022 foram reservados para comparar com o resultado dos modelos estatísticos e algoritmos de previsibilidade que foram utilizados no teste. A Figura 18 mostra o conjunto de dados utilizado.

Figura 18 - Conjunto de dados



Fonte: Própria Autora

```
path = '/content/drive/My Drive/Radiacao/'
files =
['jequitinhonha2017.xlsx', 'jequitinhonha2018.xlsx', 'jequitinhonha2019.xlsx'
, 'jequitinhonha2020.xlsx', 'jequitinhonha2021.xlsx']
dataset = []
for file in files:
    fileName = path+file
    with open(fileName) as f:
        if len(dataset) == 0:
            dataset=pd.read_excel(fileName, sheet_name=None,
index_col=1) ['Sheet1']
```

```

else:
    dataset = pd.concat([dataset, pd.read_excel(fileName,
sheet_name=None, index_col=1)['Sheet1']])
testDataset = pd.read_excel(path+'jequitinhonha2022.xlsx', sheet_name=None,
index_col=1)['Sheet1']
dataset.drop(['Unnamed: 0'], axis=1, inplace = True)
testDataset.drop(['Unnamed: 0'], axis=1, inplace = True)

```

Esse trecho do código realiza a leitura e combina os dados de data e irradiância de múltiplos arquivos Excel em um único DataFrame, através da biblioteca pandas. Logo, os dados dos cinco primeiros anos foram abertos por meio de um loop e atribuídos ao DataFrame 'dataset' e o último arquivo foi atribuído a outro DataFrame chamado 'testDataset'. As últimas duas linhas serviram para remover as colunas indesejadas.

Com o intuito de analisar o comportamento dos dados ao longo dos anos, foram utilizadas as seguintes linhas de código para plotar os dados armazenados:

```

ax = dataset['IRRADIÂNCIA (W/m²)'].plot(figsize=(16, 5))
ax.set_ylabel('IRRADIÂNCIA (W/m²)')

# Adicionar título ao gráfico
ax.set_title('Dados de Treino')

# Mostrar o gráfico
plt.show()

ax = testDataset.plot(figsize=(16, 5))
ax.set_ylabel('IRRADIÂNCIA (W/m²)')

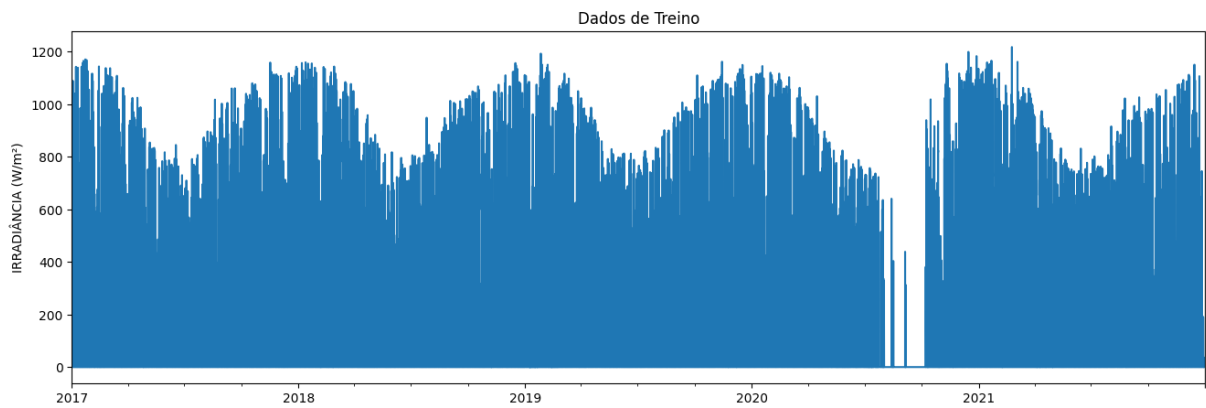
# Adicionar título ao gráfico
ax.set_title('Dados de Teste')

# Mostrar o gráfico
plt.show()

```

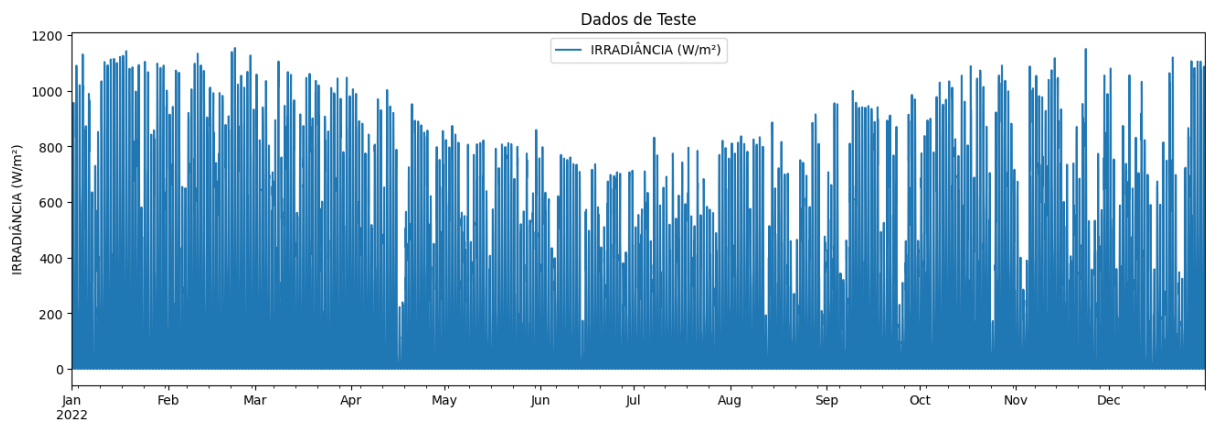
As Figura 19 e Figura 20 apresentados os dados plotados de treino e de comparação que serão utilizados posteriormente.

Figura 19 - Dados de Treino 2017 a 2021



Fonte: Própria Autora

Figura 20 - Dados para comparação dos resultados



Fonte: Própria Autora

Após essa etapa, a média da irradiância foi calculada para cada hora dentro do DataFrame, permitindo analisar se o perfil diário da curva estava conforme o esperado. Esse processo possibilitou verificar se o padrão de variação da irradiância ao longo do dia estava de acordo com as expectativas prévias.

```
ax = dataset['IRRADIÂNCIA
(W/m2)'].groupby(by=[dataset.index.hour]).mean().plot(figsize=(16, 5))
ax.set_ylabel('IRRADIÂNCIA (W/m2)')

# Adicionar título ao gráfico
ax.set_title('Média Diária dos Dados de Treino')

# Mostrar o gráfico
plt.show()
```

```

ax =
testDataset.groupby(by=[testDataset.index.hour]).mean().plot(figsize=(16,
5))
ax.set_ylabel('IRRADIÂNCIA (W/m²)')

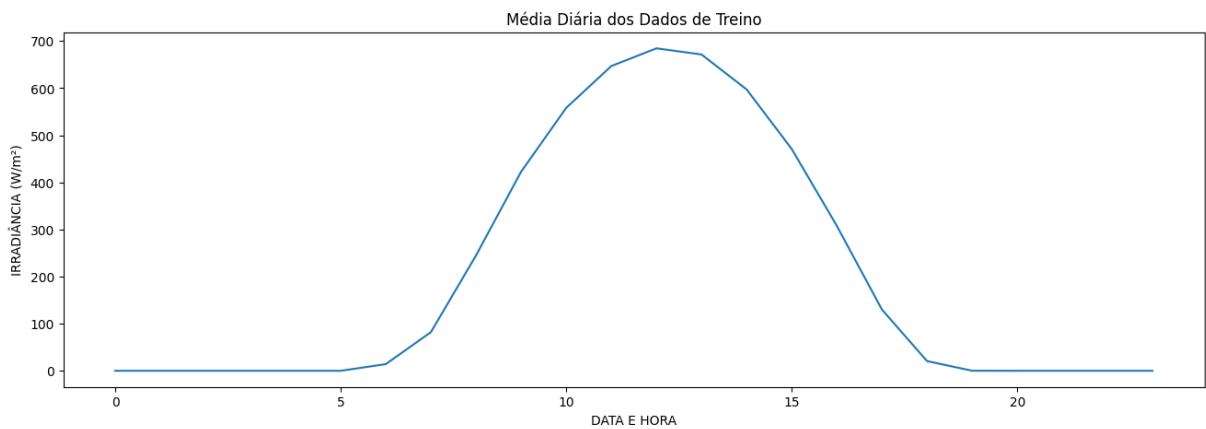
# Adicionar título ao gráfico
ax.set_title('Média Diária dos Dados de Teste')

# Mostrar o gráfico
plt.show()

```

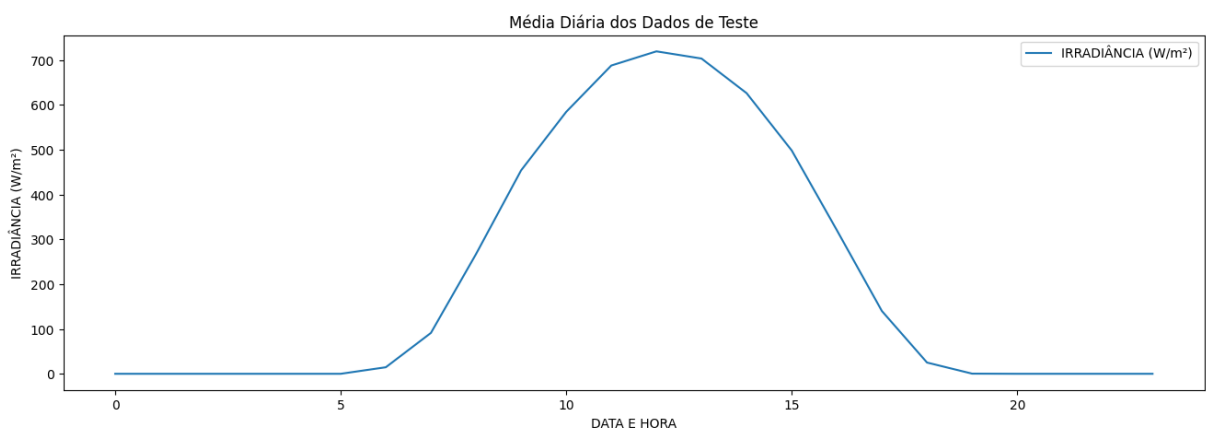
As Figura 21 e Figura 22 apresentam as medidas diárias dos dados de treino e de teste.

Figura 21 - Média diária dos dados de treino



Fonte: Própria Autora

Figura 22 - Média diária dos dados de teste



Fonte: Própria Autora

3.3.1 ARIMA

Antes de aplicar o modelo ARIMA, é necessário verificar se a série de dados é estacionária. Caso contrário, é necessário realizar etapas adicionais para transformá-la em uma série estacionária. No trecho de código a seguir, é possível utilizar uma função em Python para determinar se a série de dados é estacionária. Essa função, chamada ‘adfuller’, é um método estatístico que realiza o teste da raiz unitária. Esse trecho nos ajuda a verificar se a série de dados possui uma tendência constante ao longo do tempo, permitindo assim que ela seja modelada ou removida, caso necessário. No caso fornecido, a função ‘adfuller’ é aplicada aos dados de entrada armazenados no DataFrame ‘dataset’. O resultado retornado nesse caso são os ‘valores-p’, que indicam a probabilidade associada à estatística do teste. Essa função faz parte da biblioteca “statsmodels”.

```
result = adfuller(dataset['IRRADIÂNCIA (W/m2)'])  
result[1] # p-value
```

O Valor retornado foi: 1,090685473053738e-25.

Se o valor for menor que 0,05 significa que é estacionário, logo apresenta um padrão constante ao longo do tempo. Como o valor mencionado foi próximo de zero, é possível afirmar que é uma série estacionária.

O método ARIMA é adequado para lidar com séries univariadas que apresentam tendências e que não possuem componentes sazonais. Para analisar e compreender a estrutura sazonal de uma série, é necessário realizar a decomposição sazonal de uma série temporal usando o método da decomposição aditiva. Esse processo envolve a separação da série em seus componentes principais, como a série original, a tendência, a sazonalidade e os resíduos. Ao plotar esses componentes em quatro gráficos separados, podemos visualizar a estrutura sazonal da série e entender os seus padrões. Com isso, os trechos de códigos a seguir têm como objetivo realizar a decomposição mencionada, a fim de verificar a presença de sazonalidade. Na Figura 23 está a decomposição realizada.

```
decomposition = seasonal_decompose(dataset['IRRADIÂNCIA (W/m2)'],  
model='additive')  
trend = decomposition.trend  
seasonal = decomposition.seasonal  
residual = decomposition.resid
```



```

plt.figure(figsize=(12, 8))
plt.subplot(411)
plt.plot(dataset['IRRADIÂNCIA (W/m²)'], label='Original')
plt.legend(loc='best')
plt.title('Original Data')

plt.subplot(412)
plt.plot(trend, label='Trend')
plt.legend(loc='best')
plt.title('Trend')

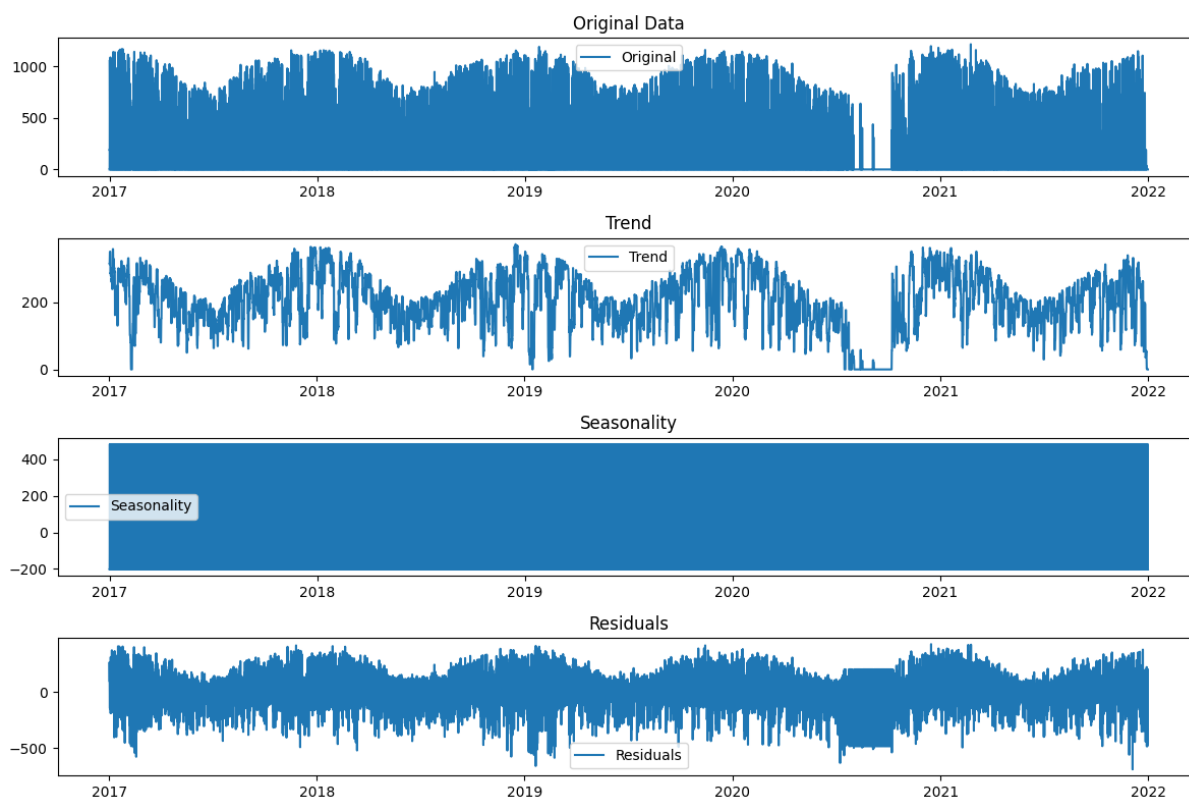
plt.subplot(413)
plt.plot(seasonal, label='Seasonality')
plt.legend(loc='best')
plt.title('Seasonality')

plt.subplot(414)
plt.plot(residual, label='Residuals')
plt.legend(loc='best')
plt.title('Residuals')

plt.tight_layout()
plt.show()

```

Figura 23 - Decomposição sazonal usando o método da decomposição aditiva



Fonte: Própria Autora

A análise da decomposição realizada mostra a presença de tendência e sazonalidade na série, no terceiro gráfico da Figura 23, indicando que o modelo ARIMA não é o mais apropriado. Nesse caso, o modelo ARIMA sazonal, também conhecido como SARIMA é mais apropriado, pois ele permite que seja adicionado uma componente sazonal na modelagem.

Com base nessa consideração, os resultados obtidos de um modelo ARIMA não são ideais, uma vez que o modelo aplicado pode não ser o mais adequado. No entanto, será apresentado um passo a passo da aplicação do modelo ARIMA sazonal.

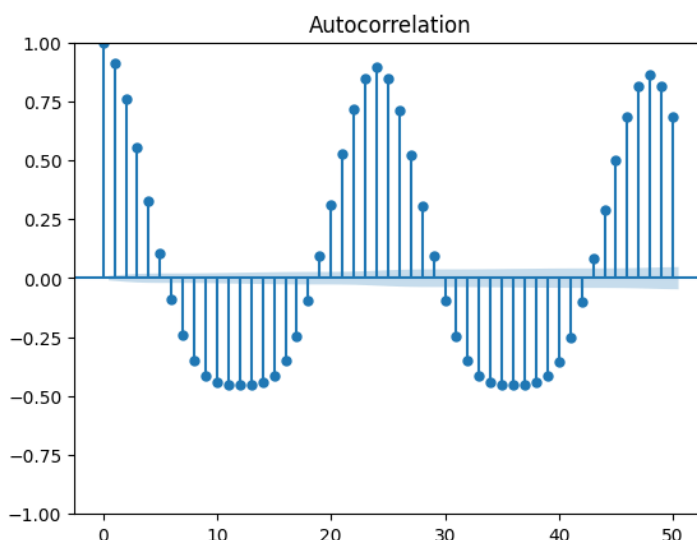
O modelo em questão é composto pela combinação de um Modelo Autorregressivo ($AR_{(p)}$) com o modelo de Dados de Médias Móveis ($MA_{(q)}$), esse modelo também utiliza a Diferenciação ($I_{(d)}$). Por meio de análises de gráficos ou códigos, são definidas as ordens dos parâmetros (p, d, q) para utilizar no modelo ARIMA.

Com base na primeira análise, foi constatado que a série é estacionária, o que implica que não é necessário realizar diferenciação. Portanto, no código, a variável 'd' será definida como 0, se fosse necessário realizar a diferenciação seria definido como 1.

Para encontrar os outros dois parâmetros, é utilizado um modelo de autocorrelação e outro de autocorrelação parcial que descrevem o comportamento da série temporal. Sendo encontrados os gráficos das Figura 24 e Figura 25 através das linhas de código a seguir.

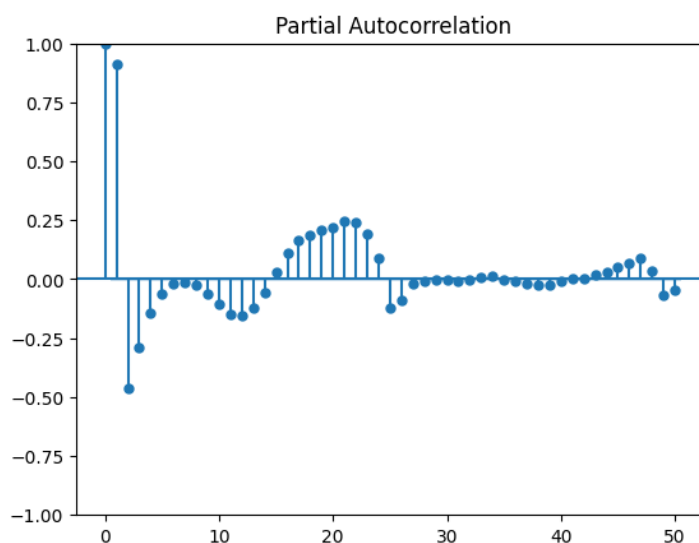
```
plot_acf(dataset['IRRADIÂNCIA (W/m²)'], lags=50)
plot_pacf(dataset['IRRADIÂNCIA (W/m²)'], lags=50)
```

Figura 24 - Gráfico de Autocorrelação



Fonte: Própria Autora

Figura 25 - Gráfico de Autocorrelação Parcial



Fonte: Própria Autora

Ao analisar os gráficos plotados de autocorrelação e autocorrelação parcial, foi observado que os valores não chegaram na faixa azul horizontal, o que dificultou a determinação direta dos parâmetros, isso pode ser explicado por conta da presença de sazonalidade e a quantidade de dados. Em casos como esse, é possível realizar tentativas e ajustes manuais para encontrar a melhor combinação desses valores, capturando os padrões da série temporal.

Com base na escolha dos parâmetros por meio de tentativa e erro, foi observado que o perfil da curva não se ajusta adequadamente sem a inclusão da componente de sazonalidade no modelo. Para incorporar essa característica, foi necessário adicionar a função de sazonalidade, onde é especificada a frequência com a qual o perfil da curva se repete nos dados. Portanto, o código ficou estruturado da seguinte maneira:

```

model = ARIMA(dataset['IRRADIÂNCIA (W/m²)'], order=(4, 0, 0),
seasonal_order=(1, 0, 1, 24))
model_fit = model.fit()

n_steps = len(testDataset['IRRADIÂNCIA (W/m²)'])
arima_predictions = pd.DataFrame()
arima_predictions.index = testDataset.index
arima_predictions['IRRADIÂNCIA (W/m²)'] =
model_fit.get_forecast(steps=n_steps).predicted_mean

```

```

arima_predictions[arima_predictions['IRRADIÂNCIA (W/m²)'] < 0] = 0

# Plotando o primeiro DataFrame
plt.plot(arima_predictions.groupby(by=[arima_predictions.index.hour]).mean(
), label='Dados Estimados')

# Plotando o segundo DataFrame
plt.plot(testDataset.groupby(by=[testDataset.index.hour]).mean(),
label='Dados Reais')

# Adicionando rótulos e títulos
plt.xlabel('Data')
plt.ylabel('IRRADIÂNCIA (W/m²)')
plt.title('Média Diária: Dados Estimados x Dados Reais')

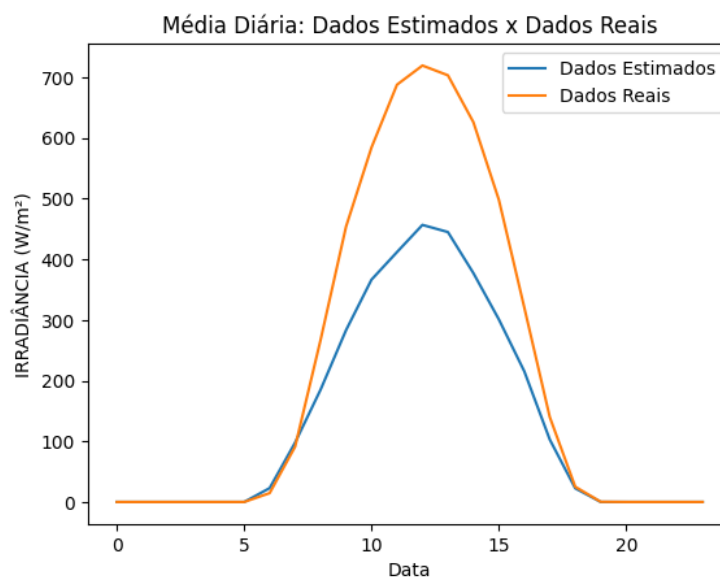
# Adicionando Legenda
plt.legend()

# Exibindo o gráfico
plt.show()

```

A média diária do modelo ARIMA sazonal está presente na Figura 26.

Figura 26 - Média diária do modelo ARIMA sazonal



Fonte: Própria Autora

Após a implementação do modelo estatístico ARIMA sazonal, foi necessário verificar a qualidade das previsões comparando-as com os valores reais. Para isso, foram selecionadas duas métricas de avaliação, o coeficiente de determinação (R^2) e o erro médio absoluto

(MAE). Através dessas métricas, é possível medir o quão bem o modelo está se ajustando aos dados e estimar a precisão das previsões. O trecho de código a seguir foi desenvolvido para calcular os valores das duas métricas de avaliação.

```
y_true = testDataset['IRRADIÂNCIA (W/m²)']
y_pred = arima_predictions['IRRADIÂNCIA (W/m²)']

# Calculo R2
r2 = r2_score(y_true, y_pred)
mae = mean_absolute_error(y_true, y_pred)

# Métricas de avaliação
print("Evaluation Metrics:")
print("-----")
print(f"R2 Score: {r2:.2f}")
print(f"MAE: {mae:.2f}")
```

Os valores retornados, foram:

$$R^2 = 0,63$$

$$MAE = 97,93$$

Os valores ideais para as métricas de avaliação são aqueles que o coeficiente de determinação (R^2) se aproxima de 1 e o erro médio absoluto (MAE) é o menor possível.

Por fim, realizou-se uma análise para identificar o número de amostras que estavam fora de um limite estabelecido em 15% do valor real. Essa avaliação teve como objetivo identificar quantas amostras do conjunto de dados apresentaram previsões que excederam o limite determinado.

```
n1 = (arima_predictions[testDataset['IRRADIÂNCIA (W/m²)']==0]['IRRADIÂNCIA
(W/m²)'] > 10).sum()

n2 = ((arima_predictions[testDataset['IRRADIÂNCIA (W/m²)']==0]['IRRADIÂNCIA
(W/m²)']-testDataset[testDataset['IRRADIÂNCIA (W/m²)']==0]['IRRADIÂNCIA
(W/m²)'])<=0.15*testDataset[testDataset['IRRADIÂNCIA
(W/m²)']==0]['IRRADIÂNCIA (W/m²)']).sum()

print('Número de amostras fora do limite: '+ str(n1+n2))
```

Número de amostras fora do limite: 3748.

3.3.2 ARIMAX

No modelo ARIMAX também são definidas variáveis, porém ela possui uma variável a mais, chamada variável exógena, podendo ser encontrada através da temperatura, pressão atmosférica ou outros dados relevantes. Como resultado, os resultados do código ARIMAX não apresentaram grandes discrepâncias em relação ao modelo estatístico anterior, ARIMA. Com isso, esse modelo também não produziu resultados satisfatórios devido à presença de sazonalidade na série temporal.

Os dados contêm duas colunas, uma para data e hora e outra para a medida de irradiância. Nesse contexto, as variáveis exógenas selecionadas são as horas do dia e o mês do ano, a fim de investigar como esses elementos podem afetar a irradiância.

Caso o modelo ARIMA não tivesse sido utilizado, seria necessário realizar os mesmos testes iniciais do modelo anterior para determinar a sua adequação. Além disso, foi necessário implementar em uma parte do código a sazonalidade, com isso, o modelo aplicado foi ARIMAX sazonal, conhecido como SARIMAX. No entanto, a seguir é apresentado o código utilizado para realizar a previsão da irradiância.

```
dataset['horas']= dataset.index.hour
dataset['mes']= dataset.index.month
exog_variables = dataset[['horas', 'mes']] # Substitua por suas váriaveis
exógenas reais
model = ARIMA(dataset['IRRADIÂNCIA (W/m²)'], order=(4, 0, 1),
seasonal_order=(1, 0, 1, 24), exog=exog_variables)
model_fit = model.fit()

n_steps = len(testDataset['IRRADIÂNCIA (W/m²)'])
arimax_predictions = pd.DataFrame()
arimax_predictions.index = testDataset.index
arimax_predictions['horas'] = arimax_predictions.index.hour
arimax_predictions['mes'] = arimax_predictions.index.month
next_year_exog = arimax_predictions[['horas', 'mes']]
arimax_predictions['IRRADIÂNCIA (W/m²)'] =
model_fit.predict(start=arimax_predictions.index[0],
end=arimax_predictions.index[-1], exog=next_year_exog)

arimax_predictions[arimax_predictions['IRRADIÂNCIA (W/m²)'] < 0] = 0

# Plotando primeiro DataFrame
```

```

plt.plot(arimax_predictions.groupby(by=[arimax_predictions.index.hour])['IRRADIÂNCIA (W/m²)'].mean(), label='Dados Estimados')

# Plotando Segundo DataFrame
plt.plot(testDataset.groupby(by=[testDataset.index.hour]).mean(), label='Dados Reais')

# Adicionando rótulos e títulos
plt.xlabel('Data')
plt.ylabel('IRRADIÂNCIA (W/m²)')
plt.title('Média Diária: Dados Estimados x Dados Reais')

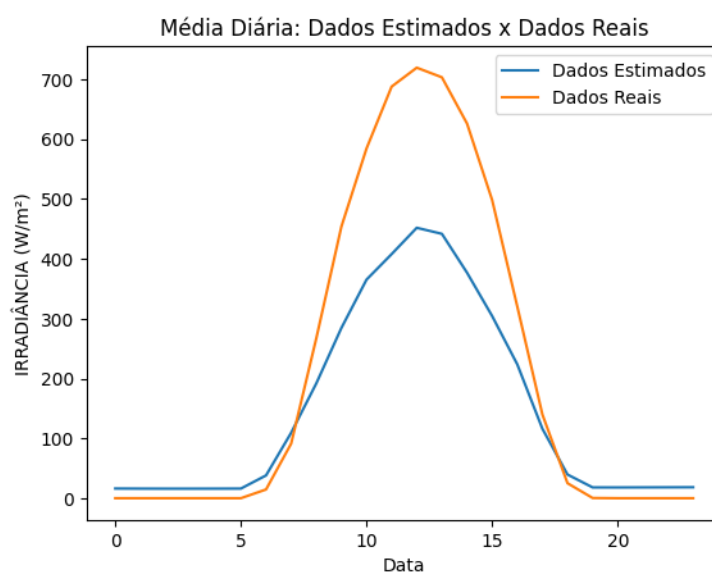
# Adicionando legenda
plt.legend()

# Exibindo o gráfico
plt.show()

```

No código acima, as variáveis exógenas são selecionadas no DataFrame “dataset” e adicionadas a “exog_variables”. O modelo é ajustado e armazenado em “model_fit”. Em seguida, é definido que todos os dados de 2017 a 2021 no “dataset” serão utilizados no teste. É criado um DataFrame para armazenar as previsões. Para obter uma curva mais suave, os valores abaixo de zero são definidos como zero. Por fim, é plotada a curva da média diária do modelo em questão. A média diária do modelo ARIMAX sazonal está presente na Figura 27.

Figura 27 - Média diária do modelo ARIMAX sazonal



Fonte: Própria Autora

Após a implementação do modelo estatístico ARIMAX sazonal, foi necessário verificar a qualidade das previsões comparando-as com os valores reais. Para isso, foram selecionadas duas métricas de avaliação, o coeficiente de determinação (R^2) e o erro médio absoluto (MAE).

Os valores retornados, foram:

$$R^2 = 0,63$$
$$MAE = 106,29$$

Os valores ideais para as métricas de avaliação são aqueles que o coeficiente de determinação (R^2) se aproxima de 1 e o erro médio absoluto (MAE) é o menor possível.

Por fim, realizou-se uma análise para identificar o número de amostras que estavam fora de um limite estabelecido em 15% do valor real.

Número de amostras fora do limite: 3874.

3.3.3 LSTM

No algoritmo de rede neural LSTM, um passo inicial importante é a normalização dos dados. Essa etapa é realizada para facilitar o manuseio dos dados e é muito relevante na implementação de algoritmos de aprendizado de máquina, onde o ajuste da escala dos dados é fundamental. Caso tivesse mais de uma variável de dados envolvida, essa etapa se tornaria ainda mais crucial. Para realizar a normalização, foi definida uma função no código que retorna os dados normalizados como resultado, assim como pode ser visto a seguir.

```
def normalize(data): #Normalizar os dados de entrada
    data_mean = data.mean(axis=0) #Média
    data_std = data.std(axis=0) #Desvio Padrão
    return (data - data_mean) / data_std
```

Com a função acima foi possível normalizar os dados de irradiância. Além disso, como os valores dependem de horas e datas específicas, foi usado uma função para converter o índice em objetos de data e hora.


```
# Etapa 1: Converter índice em objeto datetime data e hora
dataset.index = pd.to_datetime(dataset.index)

# Etapa 2: Pré-processar os dados

# Normalizar a 'IRRADIÂNCIA (W/m²)' em valores entre 0 e 1
dataset['irradiation_normalized'] = normalize(dataset['IRRADIÂNCIA
(W/m²)'])
```

É importante realizar uma preparação dos dados para o treinamento do modelo de aprendizado de máquina. Nesse contexto, uma variável “train_df” recebe os dados presentes no DataFrame “dataset”. Em seguida, é definido o comprimento desejado das sequências de entrada, que nesse caso é de 24 horas. Uma função de treinamento é definida para processar os valores de entrada (x) e os valores previstos (y), e ao final, são retornados utilizando a biblioteca numpy.

```
# Os dados serão separados em teste e treino

train_df = dataset

# Etapa 3: Preparando a entrada

sequence_length = 24 #Comprimento da sequência de entrada(24h)

def create_sequences(X, y, sequence_length):
    X_sequences = []
    y_values = []
    for i in range(len(X) - sequence_length):
        X_sequences.append(X[i:i+sequence_length])
        y_values.append(y[i+sequence_length])
    return np.array(X_sequences), np.array(y_values)

# Crie sequências de treinamento e seus valores de saída correspondente
X_train, y_train = create_sequences(train_df['irradiation_normalized'],
train_df['irradiation_normalized'], sequence_length)
```

As próximas etapas foram responsáveis por definir e treinar o modelo, onde foi definido quantas unidades de neurônios possuem na camada, quanto maior esse número melhor será para o aprendizado do algoritmo em questão. E foi definido outros parâmetros para o treinamento do modelo, como pode ser visto nas linhas de código a seguir:

```
# Etapa 4: Construir o modelo LSTM
```

```

model = Sequential()
model.add(LSTM(50, input_shape=(sequence_length, 1))) # 50 unidades na
camada LSTM
model.add(Dense(1)) # Camada de saída com 1 unidade
model.compile(loss='mean_squared_error', optimizer='adam') # Usando o erro
quadrático médio como a função de perda

# Etapa 5: Treine o Modelo LSTM

epochs = 12 # Número de épocas do treinamento
batch_size = 32 # Quantidade de treinamentos sendo realizados em paralelo
antes do peso do modelo ser ajustado
model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size)

```

Após isso, foram definidas as datas de início e término das previsões e adicionadas essas informações ao DataFrame “lstm_predictions”. Ele será usado para armazenar as previsões do modelo LSTM, junto a coluna de data e hora já adicionadas.

```

# Etapa 7: Prever os dados do próximo ano
test_year_start = pd.to_datetime('2021-12-31 00:00:00')
next_year_start = pd.to_datetime('2022-01-01 00:00:00')
next_year_end = pd.to_datetime('2022-12-31 23:00:00')

# Crie um dataframe vazio com o índice datetime para o próximo ano
test_df = pd.DataFrame(index=pd.date_range(start=test_year_start,
end=next_year_end, freq='H'))
lstm_predictions = pd.DataFrame(index=pd.date_range(start=next_year_start,
end=next_year_end, freq='H'))

```

Em seguida, com o código já treinado, foram feitas as previsões. Além disso, na última linha foi feito o processo de reverter a normalização. Com isso, foram multiplicadas as previsões pelo desvio padrão dos dados originais e adicionado a média dos dados originais. Assim como mostra a seguir.

```

test_df['IRRADIÂNCIA (W/m²)'] = dataset[dataset.index >= '2020-12-
31']['IRRADIÂNCIA (W/m²)'].values

# Normalize a 'IRRADIÂNCIA (W/m²)' valores entre 0 e 1
test_df['irradiation_normalized'] = normalize(test_df['IRRADIÂNCIA
(W/m²)'])

# Dados do próximo ano em um dataframe chamado 'test_df'
test_sequences, _ = create_sequences(test_df['irradiation_normalized'],
test_df['irradiation_normalized'], sequence_length)

```

```

next_year_predictions = model.predict(test_sequences)
lstm_predictions['IRRADIÂNCIA (W/m²)'] =
(next_year_predictions)*dataset['IRRADIÂNCIA
(W/m²)'].std(axis=0)+dataset['IRRADIÂNCIA (W/m²)'].mean(axis=0)

```

Para obter uma curva mais suave da curva média diária na Figura 28, os valores abaixo de zero são definidos como zero. Assim como pode ser visto a seguir.

```

lstm_predictions[lstm_predictions['IRRADIÂNCIA (W/m²)'] < 0] = 0
# Plotando o primeiro DataFrame
plt.plot(lstm_predictions.groupby(by=[lstm_predictions.index.hour]).mean(),
label='Dados Estimados')

# Plotando o segundo DataFrame
plt.plot(testDataset.groupby(by=[testDataset.index.hour]).mean(),
label='Dados Reais')

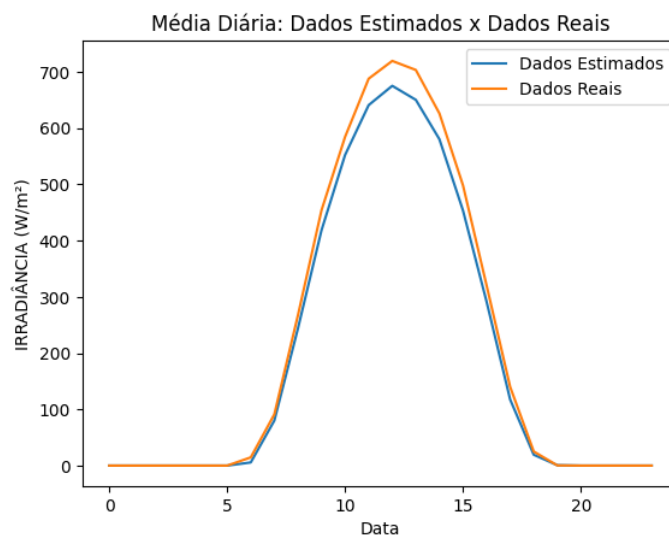
# Adicionando rótulos e título
plt.xlabel('Data')
plt.ylabel('IRRADIÂNCIA (W/m²)')
plt.title('Média Diária: Dados Estimados x Dados Reais')

# Adicionando a legenda
plt.legend()

# Exibindo o gráfico
plt.show()

```

Figura 28 - Média diária do algoritmo LSTM



Fonte: Própria Autora

Após a implementação do algoritmo de rede neural LSTM, foi necessário verificar a qualidade das previsões comparando-as com os valores reais. Para isso, foram selecionadas duas métricas de avaliação, o coeficiente de determinação (R^2) e o erro médio absoluto (MAE).

Os valores retornados, foram:

$$R^2 = 0,72$$

$$MAE = 79,11$$

Os valores ideais para as métricas de avaliação são aqueles que o coeficiente de determinação (R^2) se aproxima de 1 e o erro médio absoluto (MAE) é o menor possível.

Por fim, realizou-se uma análise para identificar o número de amostras que estavam fora de um limite estabelecido em 15% do valor real.

Número de amostras fora do limite: 3929.

3.3.4 GRU

No algoritmo de rede neural GRU, assim como no algoritmo LSTM, é realizada a normalização dos dados. Para realizar essa normalização, também foi utilizada a mesma função no código que retorna os dados normalizados como resultado, assim como pode ser visto a seguir.

```
# Normalizar a 'IRRADIÂNCIA (W/m²)' em valores entre 0 e 1
dataset['irradiation_normalized'] = normalize(dataset['IRRADIÂNCIA
(W/m²)'])
```

Na preparação dos dados para o treinamento do modelo de aprendizado de máquina atual, foi implementado as mesmas linhas de código do algoritmo anterior. Porém na etapa de definição e treinamento do modelo, foi necessário realizar uma modificação, consistindo apenas na alteração da função utilizada.

```
# Etapa 4: construir o modelo GRU
```

```

model = Sequential()
model.add(GRU(units=64, input_shape=(sequence_length, 1)))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')

# Etapa 5: Treinar o modelo GRU

epochs = 12
batch_size = 32
model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size)

```

Em seguida, com o código já treinado, foram feitas as previsões. Além disso, na última linha foi feito o processo de reverter a normalização. Com isso, foram multiplicadas as previsões pelo desvio padrão dos dados originais e adicionado a média dos dados originais. Assim como mostra a seguir.

```

# Etapa 7: prever os dados do próximo ano
test_year_start = pd.to_datetime('2021-12-31 00:00:00')
next_year_start = pd.to_datetime('2022-01-01 00:00:00')
next_year_end = pd.to_datetime('2022-12-31 23:00:00')

# Crie um dataframe vazio com o índice datetime para o próximo ano
test_df = pd.DataFrame(index=pd.date_range(start=test_year_start,
end=next_year_end, freq='H'))

gru_predictions = pd.DataFrame(index=pd.date_range(start=next_year_start,
end=next_year_end, freq='H'))

test_df['IRRADIÂNCIA (W/m²)'] = dataset[dataset.index >= '2020-12-31']
['IRRADIÂNCIA (W/m²)'].values

# Normalizar a 'IRRADIÂNCIA (W/m²)' em valores entre 0 e 1
test_df['irradiation_normalized'] = normalize(test_df['IRRADIÂNCIA
(W/m²)'])

# Dados do próximo ano em um dataframe chamado 'test_df'
test_sequences = create_sequences(test_df['irradiation_normalized'],
test_df['irradiation_normalized'], sequence_length)

next_year_predictions = model.predict(test_sequences)

gru_predictions['IRRADIÂNCIA (W/m²)'] =
(next_year_predictions)*dataset['IRRADIÂNCIA
(W/m²)'].std(axis=0)+dataset['IRRADIÂNCIA (W/m²)'].mean(axis=0)

```

Para obter a curva média diária, foi plotado o gráfico da Figura 29 utilizando algumas linhas de código. Assim como pode ser visto a seguir.

```
gru_predictions[gru_predictions['IRRADIÂNCIA (W/m²)'] < 0] = 0
# Plotando o primeiro DataFrame
plt.plot(gru_predictions.groupby(by=[gru_predictions.index.hour]).mean(),
label='Dados Estimados')

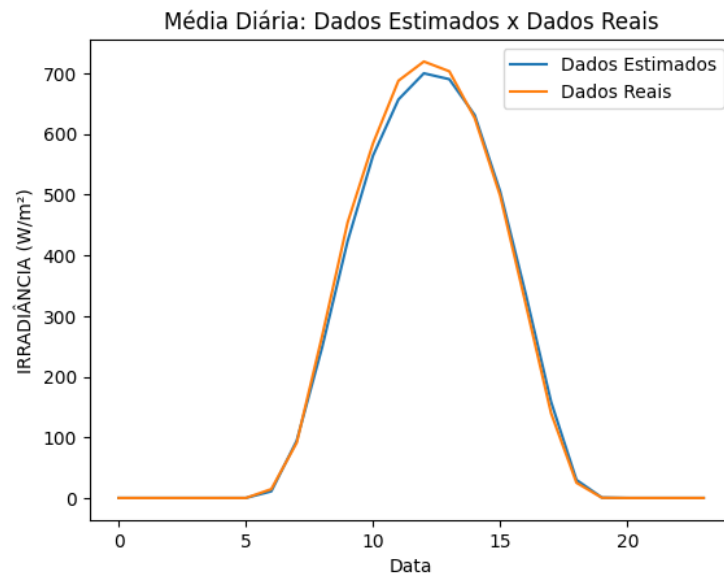
# Plotando o segundo DataFrame
plt.plot(testDataset.groupby(by=[testDataset.index.hour]).mean(),
label='Dados Reais')

# Adicionando rótulos e título
plt.xlabel('Data')
plt.ylabel('IRRADIÂNCIA (W/m²)')
plt.title('Média Diária: Dados Estimados x Dados Reais')

# Adicionando legenda
plt.legend()

# Exibindo o gráfico
plt.show()
```

Figura 29 - Média diária do algoritmo GRU



Fonte: Própria Autora

Após a implementação do algoritmo de rede neural GRU, foi necessário verificar a qualidade das previsões comparando-as com os valores reais. Para isso, foram selecionadas

duas métricas de avaliação, o coeficiente de determinação (R^2) e o erro médio absoluto (MAE).

Os valores retornados, foram:

$$R^2 = 0,72$$
$$MAE = 79,34$$

Os valores ideais para as métricas de avaliação são aqueles que o coeficiente de determinação (R^2) se aproxima de 1 e o erro médio absoluto (MAE) é o menor possível.

Por fim, realizou-se uma análise para identificar o número de amostras que estavam fora de um limite estabelecido em 15% do valor real.

Número de amostras fora do limite: 3928.

3.3.5 CNN

O primeiro passo no algoritmo de rede neural CNN, assim como nos outros dois algoritmos de aprendizado de máquina e realizar a normalização dos dados. Para realizar essa normalização, também foi utilizada a mesma função criada anteriormente que retorna os dados normalizados como resultado, assim como pode ser visto a seguir.

```
data = normalize(dataset['IRRADIÂNCIA (W/m2)'].values)
```

A preparação adequada dos dados é essencial para o treinamento de um modelo de aprendizado de máquina. Nesse contexto, foram necessárias algumas etapas para preparar os dados antes do treinamento. Primeiro, definiu-se o número de horas anteriores a serem consideradas na previsão, o que é importante para capturar os padrões temporais nos dados. Em seguida, as variáveis de entrada (x) e a variável de destino (y) foram preparadas e convertidas em matrizes Numpy para facilitar o processamento. Porém, a rede neural em questão espera que os dados tenham um formato 3D. Com isso, os dados de entrada (x) foram remodelados, onde na primeira camada se encontra o número de amostras, na segunda o número de passos anteriores que vão ser considerados para prever o próximo e a última camada é o número de variáveis de entrada.

```

# Defina o número de horas anteriores a considerar para previsão
num_timesteps = 24

# Preparar variáveis de entrada e de destino
X = []
y = []
for i in range(num_timesteps, len(data)):
    X.append(data[i-num_timesteps:i])
    y.append(data[i])

# Converter em matrizes numpy
X = np.array(X)
y = np.array(y)

# Remodelar dados de entrada para array 3D
X = X.reshape(X.shape[0], X.shape[1], 1)

```

As próximas etapas foram responsáveis por definir e treinar o modelo, onde são foram definidas camadas para extrair as características e os padrões dos dados. Assim como pode ser visto nas linhas de código a seguir:

```

# Divida os dados em conjuntos de treinamento e teste
X_train = X
y_train=y
# Criar um modelo sequencial
model = Sequential()

# Adicionar camada convolucional com 32 filtros, tamanho de kernel 3 e
ativação Relu
model.add(Conv1D(32, 3, activation='relu', input_shape=(num_timesteps, 1)))

# Adicionar camada de pool máximo com tamanho de pool 2
model.add(MaxPooling1D(pool_size=2))

# Adicione outra camada convolucional com 64 filtros e tamanho de kernel 3
model.add(Conv1D(64, 3, activation='relu'))

# Adicione outra camada de pool máximo
model.add(MaxPooling1D(pool_size=2))

# Achatar a saída da camada anterior
model.add(Flatten())

# Adicione uma camada totalmente conectada com 128 unidades e ativação ReLU
model.add(Dense(128, activation='relu'))

```



```

# Adicione uma camada de saída com uma única unidade (regressão) e ativação
linear
model.add(Dense(1, activation='linear'))

# Compile o modelo com perda de erro ao quadrado médio e otimizador de Adam
model.compile(loss='mean_squared_error', optimizer='adam')

# Treinando o modelo
model.fit(X_train, y_train, epochs=15, batch_size=32)

```

Em seguida, com o código já treinado, foram feitas as previsões. Além disso, na última linha foi feito o processo de reverter a normalização. Assim como mostra a seguir.

```

# Faça previsões sobre novos dados (irradiação solar do próximo ano)
next_year_start = pd.to_datetime('2022-01-01 00:00:00')
next_year_end = pd.to_datetime('2022-12-31 23:00:00')

size = 24*366
next_year_data = data[-1*size:]
X_test = []
for i in range(num_timesteps, len(next_year_data)):
    X_test.append(next_year_data[i-num_timesteps:i])
X_test = np.array(X_test)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
# Gerar previsões
predictions = model.predict(X_test)
cnn_predictions = pd.DataFrame(index=pd.date_range(start=next_year_start,
end=next_year_end, freq='H'))
cnn_predictions['IRRADIÂNCIA (W/m²)'] = (predictions)*dataset['IRRADIÂNCIA
(W/m²)'].std(axis=0)+dataset['IRRADIÂNCIA (W/m²)'].mean(axis=0)

```

Para obter a curva média diária da Figura 30, foi plotado o gráfico utilizando algumas linhas de código. Assim como pode ser visto a seguir.

```

# Plotando o primeiro DataFrame
plt.plot(cnn_predictions.groupby(by=[cnn_predictions.index.hour]).mean(),
label='Dados Estimados')

# Plotando o segundo DataFrame
plt.plot(testDataset.groupby(by=[testDataset.index.hour]).mean(),
label='Dados Reais')

# Adicionando rótulos e título
plt.xlabel('Data')
plt.ylabel('IRRADIÂNCIA (W/m²)')

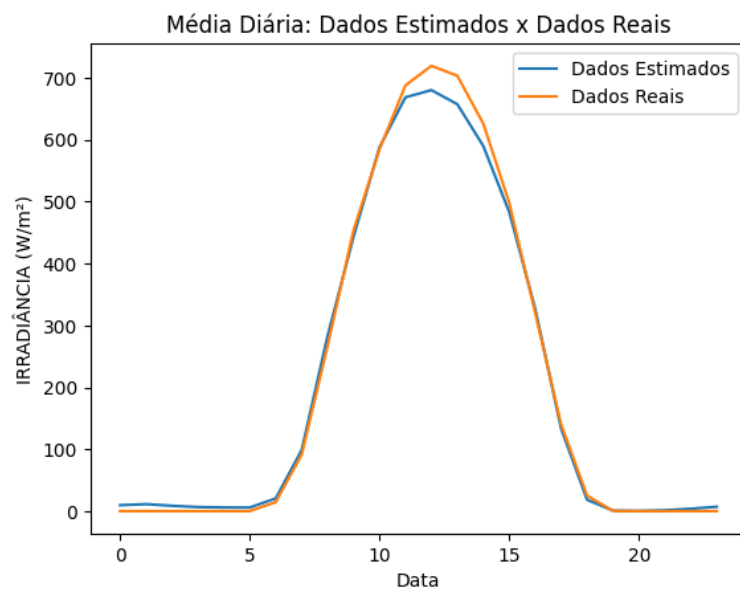
```

```
plt.title('Média Diária: Dados Estimados x Dados Reais')

# Adicionando legenda
plt.legend()

# Exibindo o gráfico
plt.show()
```

Figura 30 - Média diária do algoritmo CNN



Fonte: Própria Autora

Após a implementação do algoritmo de rede neural CNN, foi necessário verificar a qualidade das previsões comparando-as com os valores reais. Para isso, foram selecionadas duas métricas de avaliação, o coeficiente de determinação (R^2) e o erro médio absoluto (MAE).

Os valores retornados, foram:

$$R^2 = 0,75$$

$$MAE = 77,34$$

Os valores ideais para as métricas de avaliação são aqueles que o coeficiente de determinação (R^2) se aproxima de 1 e o erro médio absoluto (MAE) é o menor possível.

Por fim, realizou-se uma análise para identificar o número de amostras que estavam fora de um limite estabelecido em 15% do valor real.

Número de amostras fora do limite: 2041.

3.4 COMPARAÇÃO DOS RESULTADOS

Esse trabalho de conclusão de curso buscou verificar a eficácia dos algoritmos, por meio de comparações entre os dados previstos nos modelos estáticos e os algoritmos de previsibilidade de rede neural. Com as comparações realizadas, foi possível identificar qual modelo se aplica melhor, visando seu emprego na análise de complementaridade de fontes no contexto da implementação de sistemas híbridos de energia renovável. Inicialmente, foi plotado um gráfico na Figura 31 com os resultados compilados junto aos dados reais, assim como pode ser visto a seguir.

```
# Gerar grafico medio diario com todos resultados
# Plotar primeiro DataFrame
plt.plot(arima_predictions.groupby(by=[arima_predictions.index.hour]).mean(
), label='ARIMA Sazonal')
plt.plot(arimax_predictions.groupby(by=[arimax_predictions.index.hour])['IR
RADIÂNCIA (W/m²)'].mean(), label='ARIMAX Sazonal')
plt.plot(lstm_predictions.groupby(by=[lstm_predictions.index.hour]).mean(),
label='LSTM')
plt.plot(gru_predictions.groupby(by=[gru_predictions.index.hour]).mean(),
label='GRU')
plt.plot(cnn_predictions.groupby(by=[cnn_predictions.index.hour]).mean(),
label='CNN')

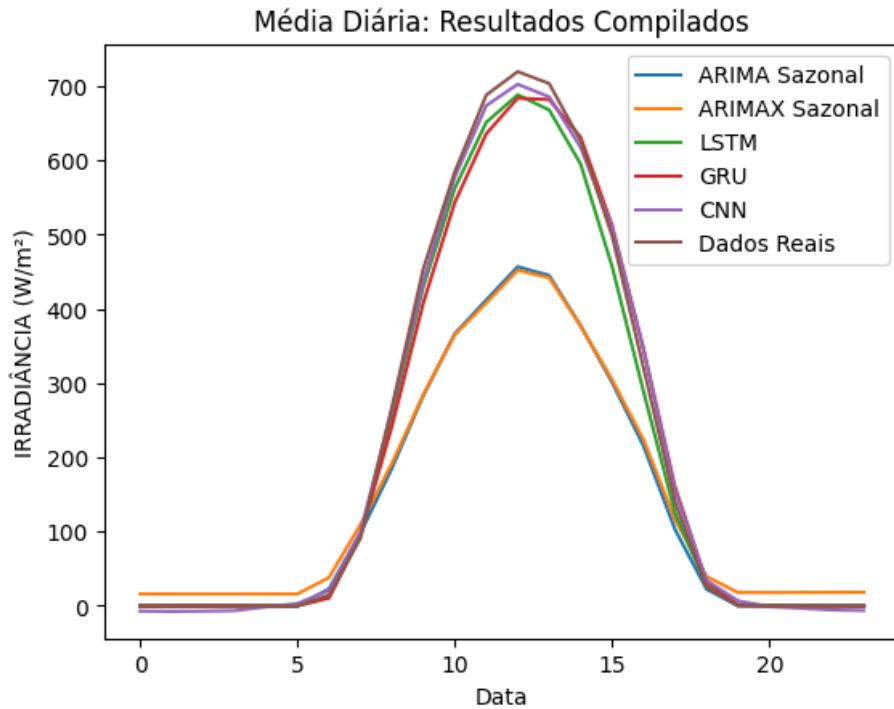
# Plotar segundo DataFrame
plt.plot(testDataset.groupby(by=[testDataset.index.hour]).mean(),
label='Dados Reais')

# Adicionando rótulos e títulos
plt.xlabel('Data')
plt.ylabel('IRRADIÂNCIA (W/m²)')
plt.title('Média Diária: Resultados Compilados')

# Adicionando legenda
plt.legend()

# Exibindo o gráfico
plt.show()
```

Figura 31 - Comparação da média diária dos modelos e algoritmos



De acordo com os gráficos, o perfil da média diária dos modelos estatísticos ARIMA sazonal e ARIMAX sazonal não ficaram próximos aos dados reais. Por outro lado, os demais algoritmos apresentaram um perfil da curva semelhante ao esperado. Porém, para se ter uma melhor análise é necessário estabelecer métricas de avaliação, no tópico passado foram implementados dois métodos, sendo eles o coeficiente de determinação (R^2) e o erro médio absoluto (MAE). A Tabela 1 e 2 apresentam esses resultados.

Tabela 1 – Métricas de avaliação

MODELOS	R^2	MAE
ARIMA sazonal	0,63	97,93
ARIMAX sazonal	0,63	106,29
LSTM	0,72	79,11
GRU	0,72	79,34
CNN	0,75	77,34

Fonte: Própria Autora

Por fim, foi realiza uma análise em cada modelo para identificar o número de amostras que estavam fora de um limite estabelecido em 15% do valor real.

Tabela 2 – Amostras fora do limite de 15%

MODELOS	Fora do limite	% Fora do limite
ARIMA sazonal	3748	42,78%
ARIMAX sazonal	3874	44,22%
LSTM	3929	44,85%
GRU	3928	44,84%
CNN	2041	23,30%

Fonte: Própria Autora

4 CONCLUSÃO

Esse trabalho de conclusão de curso explorou a viabilidade de combinar duas fontes renováveis como uma forma de contribuir para a redução das emissões de carbono. Com o objetivo de garantir uma geração de energia elétrica mais estável, foram conduzidas pesquisas sobre sistemas híbridos, que integram uma usina fotovoltaica e uma usina hidrelétrica já existente. No entanto, para um planejamento adequado e para determinar se a criação de uma usina híbrida é vantajosa, é necessário avaliar a capacidade de geração de energia da fonte que será adicionada. Portanto, foi fundamental estudar algoritmos de previsão e modelos estatísticos para identificar o melhor modelo para a realização dessas previsões, a fim de auxiliar na complementariedade.

A implementação dos modelos ARIMA sazonal e ARIMAX sazonal foram as mais desafiadoras devido à necessidade de definir os parâmetros por meio de gráficos. Embora a opção de usar a biblioteca que contenha o método autoarima estivesse disponível para determinar os melhores parâmetros, ela não funcionou corretamente devido à grande quantidade de dados de entrada na máquina utilizada. Portanto, os valores foram ajustados manualmente. No caso dos algoritmos de aprendizado de máquina, os desafios encontrados estavam relacionados ao treinamento do modelo.

Ao analisar os resultados dos modelos estatísticos, verificou-se que não houve uma diferença significativa entre os modelos ARIMA e ARIMAX. Isso ocorreu, pois, a única diferença entre os modelos foi a inclusão de variáveis exógenas, sendo que as variáveis utilizadas foram as horas e a data, as quais têm um impacto mínimo nos resultados. Como resultado, o desempenho do ARIMAX foi ligeiramente inferior em comparação ao ARIMA.

Ao examinar os resultados obtidos pelos algoritmos de rede neural, constatou-se que os modelos LSTM e GRU produziram resultados semelhantes. Isso se deve ao fato de que ambos os modelos são bastante similares. No entanto, o LSTM apresentou um desempenho levemente superior, o que pode ser atribuído ao fato de ter a camada a mais, e a execução do GRU foi mais rápida. Por outro lado, o modelo CNN possui uma estrutura distinta dos demais e apresentou resultados superiores em comparação aos demais.

Conclui-se, portanto, que dentre os cinco modelos estudados, o CNN se aplicou melhor para essa característica de dados. Porém, de acordo com a literatura os algoritmos LSTM e GRU são amplamente utilizados se tratando de previsões temporais, e os resultados de ambos não foram muito diferentes do CNN. Com isso, visando seu emprego na análise de complementariedade de fontes no contexto da implementação de sistemas híbridos de energia

renovável, os três modelos apresentaram resultados satisfatórios. E nos modelos estatísticos é preciso ter um cuidado ao escolher os parâmetros, o que pode dificultar ou até mesmo não trazer resultados tão satisfatórios.

Para trabalhos futuros, tanto a autora quanto outros pesquisadores podem explorar diversas possibilidades adicionais que não foram aqui contempladas. Uma abordagem interessante seria testar uma variedade algoritmos de previsão, além dos que foram utilizados neste estudo, a fim de identificar um método mais apropriado para a análise de previsibilidade com vista a verificação de complementariedade da usina híbrida. Diferentes algoritmos, como redes neurais, árvores de decisão, regressão linear, entre outros, poderiam ser aplicados e comparados com base em critérios de desempenho específicos.

Além disso, com os dados gerados pelo estudo, é possível realizar uma análise mais aprofundada da complementariedade entre a usina fotovoltaica e a usina hidrelétrica existente. Essa análise poderia incluir a avaliação do comportamento conjunto das duas fontes de energia em diferentes períodos. Isso permitiria uma compreensão mais completa do potencial de geração da usina com as duas fontes. Em resumo, há um vasto campo de trabalho futuro que pode ser explorado para aprimorar o estudo de complementariedade da usina hidrelétrica e sua contribuição para a descarbonização.

REFERÊNCIAS

- [1] EPE. **Plano decenal de expansão de energia 2023.**
- [2] ONS. **Evolução da Capacidade Instalada.** Disponível em: <https://www.ons.org.br/paginas/sobre-o-sin/o-sistema-em-numeros>. Acesso em 20 mar. 2023.
- [3] MINISTÉRIO DE MINAS E ENERGIA. **Matriz elétrica brasileira apresenta expansão de 2.746 MW no primeiro trimestre de 2023.** Disponível em: <https://www.gov.br/mme/pt-br/assuntos/noticias/matriz-eletrica-brasileira-apresenta-expansao-de-2-746-mw-no-primeiro-trimestre-de-2023>. Acesso em: 16 abr. 2023.
- [4] ANEEL. Resolução normativa n° 954, de 30 de novembro de 2021. altera a resolução normativa n° 77, de 21 de dezembro de 2006, n° 559 de 27 de junho de 2013, n° 583, de 22 de outubro de 2013, n° 666, de 23 de junho de 2015 e n° 876, de março de 2020 – prodist. Diário Oficial da União (dez. 2021).
- [5] CANAL SOLAR. **GD solar tem o melhor início de ano da história do setor.** Disponível em: <https://canalsolar.com.br/gd-solar-tem-melhor-inicio-de-ano-da-historia-do-setor/>. Acesso em: 13 jun. 2023.
- [6] CANAL SOLAR. **Brasil iniciou 2023 com redução de 20% no custo dos módulos FV.** Disponível em: [https://canalsolar.com.br/brasil-iniciou-2023-com-reducao-de-20-no-custo-dos-modulos-fv/#:~:text=Brasil%20iniciou%202023%20com%20redu%C3%A7%C3%A3o%20de%2020%25%20no%20custo%20dos%20m%C3%B3dulos%20FV,-Custo%20dos%20m%C3%B3dulos&text=O%20pre%C3%A7o%20de%20custo%20dos,\(28\)%20pela%20Greener](https://canalsolar.com.br/brasil-iniciou-2023-com-reducao-de-20-no-custo-dos-modulos-fv/#:~:text=Brasil%20iniciou%202023%20com%20redu%C3%A7%C3%A3o%20de%2020%25%20no%20custo%20dos%20m%C3%B3dulos%20FV,-Custo%20dos%20m%C3%B3dulos&text=O%20pre%C3%A7o%20de%20custo%20dos,(28)%20pela%20Greener). Acesso em: 13 jun. 2023.
- [7] NEOENERGIA. **Descarbonização.** Disponível em: <https://www.neoenergia.com/pt-br/mobilidade-eletrica/Paginas/descarbonizacao.aspx>. Acesso em 21 mar. 2023.

[8] SOLARVOLT. **Descarbonização do ar: Entenda o papel da energia solar na redução de CO2 na atmosfera.** Disponível em: <https://www.solarvoltenergia.com.br/blog/descarbonizacao-energia-solar/>. Acesso em: 3 abr. 2023.

[9] CANAL ENERGIA. **Aneel publica resolução sobre usinas híbridas e associadas.** Disponível em: <https://www.canalenergia.com.br/noticias/53195762/aneel-publica-resolucao-sobre-usinas-hibridas-e-associadas>. Acesso em: 6 abr. 2023.

[10] EPE. **Estudos de Planejamento da Expansão da Geração.** Disponível em: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/estudos-de-planejamento-da-expansao-da-geracao>. Acesso em 02 mar. 2023.

[11] HEADENERGIA. **Usinas Híbridas e associadas: Conceitos e desafios.** Disponível em: <https://www.headenergia.net/post/usinas-h%C3%ADbridas-e-associadas-conceitos-e-desafios>. Acesso em: 20 abr. 2023.

[12] NETZERO. **O que são usinas híbridas?** Disponível em: <https://netzero.projetodraft.com/o-que-sao-usinas-hibridas/>. Acesso em: 6 abr. 2023.

[13] ALDOBLOG. **Energia Solar nas hidrelétricas: aumento de 17% na geração de energia.** Disponível em: <https://www.aldo.com.br/blog/energia-solar-nas-hidreletricas-aumento-de-17-na-geracao-de-energia/>. Acesso em: 9 abr. 2023.

[14] PORTAL BLOSSISTEMAS. **Usina solar flutuando, e o Brasil.** Disponível em: usp.br/portalblossistemas/?p=8131. Acesso em 10 abril 2023.

[15] CENÁRIOS SOLAR. **Os revezes legais e o desafio de tornar o Brasil a matriz energética mais limpa do mundo.** Disponível em: <https://cenariosolar.editorabrasilenergia.com.br/os-revezes-legais-e-o-desafio-de-tornar-o-brasil-a-matriz-energetica-mais-limpa-do-mundo/>. Acesso em: 10 abr. 2023.

- [16] Brasil Escola. **Matriz energética brasileira.** Disponível em: <https://brasilecola.uol.com.br/brasil/matriz-energetica-brasileira.htm>. Acesso em: 6 abr. 2023.
- [17] WIKIPÉDIA. **Usina Hidrelétrica.** Disponível em: https://pt.wikipedia.org/wiki/Usina_hidrel%C3%A9trica. Acesso em: 12 abr. 2023.
- [18] PENSAMENTO VERDE. **O que é uma usina fio d'água?.** Disponível em: <https://www.pensamentoverde.com.br/economia-verde/usina-fio-dagua/>. Acesso em: 19 abr. 2023.
- [19] BLOGSPOT. **Usina Baixo Iguaçu.** Disponível em: <http://usinabaixoiguacu.blogspot.com/2014/02/usinas-hidreletricas-fio-dagua.html>. Acesso em: 12 abr. 2023.
- [20] CBIE. **O que são hidrelétricas reversíveis?.** Disponível em: <https://cbie.com.br/o-que-sao-hidroeletricas-reversiveis/>. Acesso em: 19 abr. 2023.
- [21] SENGEPI. **Conheça as usinas hidrelétricas reversíveis.** Disponível em: <http://www.senge-pi.org.br/artigo/conheca-as-usinas-hidreletricas-reversiveis-uhrs>. Acesso em: 12 abr. 2023.
- [22] CCEE. **Conceitos de Preço.** Disponível em: <https://www.ccee.org.br/precos/conceitos-precos>. Acesso em: 18 abr. 2023.
- [23] SOLARFY. **Potencial solar no Brasil.** Disponível em: <https://solarfy.com.br/blog/potencial-solar-no-brasil-confira-aqui-o-mapa-solarimetrico/>.
- [24] ALDOBLOG. **Como fazer o cálculo do sistema fotovoltaico com eficiência?.** Disponível em: <https://www.aldo.com.br/blog/como-fazer-o-calculo-do-sistema-fotovoltaico-com-eficiencia/>. Acesso em: 20 abr. 2023.
- [25] PORTAL SOLAR. **A Energia Solar no Mundo.** Disponível em: <https://www.portalsolar.com.br/energia-solar-no-mundo>. Acesso em: 16 abr. 2023.

- [26] TERRA. **Estimativa de Radiação e a Geração Solar**. Disponível em: <https://www.terra.com.br/planeta/energia/estimativa-de-radiacao-e-a-geracao-solar,021e976854ec40292708a920ed23fe5cnllowatw.html>. Acesso em: 19 abr. 2023.
- [27] ESTAT JUNIOR. **O que é um modelo de previsão?**. Disponível em: <https://estatjr.com.br/2019/08/11/o-que-e-um-modelo-de-previsao/>. Acesso em: 24 abr. 2023.
- [28] CS, Mateus; KS, Raphael. **Aplicação de métodos de previsão de demanda por séries temporais baseado em dados de compras em uma empresa do segmento fotográfico**. Trabalho de Conclusão de Curso, Universidade Federal de Fluminense, Volta Redonda, 2019.
- [29] JC, Roger. **Previsão de séries temporais com aplicações a séries de consumo de energia elétrica**. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, 2008.
- [30] YJ, Esther. **Estruturas de vizinhanças espaciais nos modelos autorregressivos e de médias móveis espaço-temporais STARMA**. Dissertação de Mestrado, Universidade de São Paulo, São Paulo, 2017.
- [31] S, Luiza. **Análise da aplicação do modelo ARIMA: estudos em uma instituição federal de ensino superior**. Trabalho de Conclusão de Curso, Universidade Federal de Santa Maria, Santa Maria, 2017.
- [32] Z, Zhang; JC, Moore. **Mathematical na Physical Fundamental of Climate Change**. 2015.
- [33] AS, Marco. **Métodos de previsão Auto-Regressivo aplicado a uma série de volume de produção de caminhões**. Trabalho de Conclusão de Curso, Universidade Estadual Paulista, Guaratinguetá, 2017.
- [34] MONOLITO NIMBUS. **Modelo ARIMA**. Disponível em: <https://www.monolitonimbus.com.br/modelo-arima/>. Acesso em: 20 abr. 2023.
- [35] NC, Henrique; SL, Paulo; BVLJ, João; CMC, Paulo. **Proposta para previsão de velocidade do vento através de modelagem híbrida elaborada a partir dos modelos**

ARIMAX e **RNA**. Disponível em: <https://www.scielo.br/j/rbmet/a/n6cRPrvLYyhdhYyJhgKFcXv/>. Acesso em: 22 abr. 2023.

[36] MB, Marcela. **Aplicação de Modelos Preditivos para a Geração de Energia Solar Fotovoltaica**. Trabalho de Conclusão do Curso. Universidade Federal de Santa Catarina, Florianópolis, 2022.

[37] SF, Thiago; GR Rafael. **Estudo Comparativo de Arquiteturas de Redes Neurais em Análise de Sentimentos**. Artigo, Universidade Federal de Mato Grosso do Sul, Mato Grosso do Sul, 2020.

[38] TA, Karine. **Previsibilidade de geração fotovoltaica em função da sujidade local**. Dissertação de Mestrado, Instituto Feral de Espírito Santo, Vitória, 2021.

[39] AC, Bruno. Previsão intra-diária de geração fotovoltaica usando redes neurais recorrentes do tipo LSTM e dados históricos de energia. Dissertação de Doutorado, Universidade Estadual Paulista, Bauru, 2021.

[40] AKSHAY SOOD. **Long Short-Term Memory**. Disponível em: <https://pages.cs.wisc.edu/~shavlik/cs638/lectureNotes/Long%20Short-Term%20Memory%20Networks.pdf>. Acesso em: 13 mai. 2023.

[41] DEEP LEARNING BOOK. **Arquitetura de redes neurais Gated Recurrent Unit (GRU)**. Disponível em: <https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-gated-recurrent-unit-gru/>. Acesso em: 28 de abr. 2023.

[42] MUNDO PROJETADO. Rede neural convolucional (CNN) – o que é e como funciona. Disponível em: <https://mundoprojetado.com.br/rede-neural-convolucional/>. Acesso em: 01 mai. 2023.

[43] UFPR. **Redes Neurais Convolucionais**. Disponível em: <https://www.ccee.org.br/precos/conceitos-precos>. Acesso em: 16 mai. 2023.

- [44] AR, Diego. **Deep learning e redes neurais convolucionais: reconhecimento automático de caracteres em placas de licenciamento automotivo.** Trabalho de Conclusão de Curso, Universidade Federal de Paraíba, João Pessoa, 2018.
- [45] MB, Marcela. **Aplicação de Modelos Preditivos para a Geração de Energia Solar Fotovoltaica.** Trabalho de Conclusão do Curso. Universidade Federal de Santa Catarina, Florianópolis, 2022.
- [46] TA, Karine. **Previsibilidade de geração fotovoltaica em função da sujidade local.** Dissertação de Mestrado, Instituto Feral de Espírito Santo, Vitória, 2021.
- [47] ISF, Wesley. **Um estudo comparativo de técnicas de detecção de outliers no contexto de classificação de dados.** Universidade do Estado do Rio Grande do Norte e Universidade Federal Rural do Semi-árido, 2019.
- [48] G. F. De Lima and A. O. Salazar. **Proposta para Identificação e Substituição de Outliers em Dados de Velocidade de Pipeline Inspection Gauges.** 2020.