

Previsão de Irradiação Solar Usando Redes Neurais Recorrentes com L.S.T.M.

João Vitor Cesario Fernandes
Curso de Engenharia Elétrica
Universidade Federal do Triângulo Mineiro
Uberaba – MG, Brasil
d202210285@uftm.edu.br

Arnaldo J. P. Rosentino
Departamento de Engenharia Elétrica
Universidade Federal do Triângulo Mineiro
Uberaba – MG, Brasil
arnaldo.rosentino@uftm.edu.br

Marcus V. B. Mendonça
Departamento de Engenharia Elétrica
Universidade Federal do Triângulo Mineiro
Uberaba – MG, Brasil
marcus.mendonca@uftm.edu.br

Elder Vicente Sobrinho
Departamento de Engenharia Elétrica
Universidade Federal do Triângulo Mineiro
Uberaba – MG, Brasil
elder.sobrinho@uftm.edu.br

Madeleine R. M. C. Albertini
Departamento de Engenharia Elétrica
Universidade Federal do Triângulo Mineiro
Uberaba – MG, Brasil
madeleine.albertini@uftm.edu.br

Fabricio A. M. Moura
Departamento de Engenharia Elétrica
Universidade Federal do Triângulo Mineiro
Uberaba – MG, Brasil
fabricio.moura@uftm.edu.br

Resumo: A crescente revolução tecnológica proporcionada pelas inteligências artificiais (IAs) tem impactado profundamente diversos setores, incluindo o segmento de energia renovável. A busca por soluções que aprimorem a produção eficiente de energia limpa impulsiona a aplicação de métodos preditivos baseados em aprendizado de máquina. Nesse contexto, a previsão precisa da irradiação solar é fundamental para otimizar o desempenho de sistemas fotovoltaicos e automatizar a operação de microgrids. Este trabalho explora a aplicação de redes neurais recorrentes (RNNs) com células Long Short-Term Memory (LSTM) para prever a irradiação solar a partir de dados históricos e condições meteorológicas. A metodologia apresentada oferece uma solução promissora para melhorar a confiabilidade e eficiência da integração de fontes renováveis nos sistemas elétricos, contribuindo para um cenário energético mais sustentável. Os resultados obtidos demonstram o potencial das redes LSTM na previsão de irradiação solar e destacam a relevância de soluções baseadas em IA para a transição energética em curso, alinhadas ao avanço de tecnologias para microgeração distribuída e à crescente demanda por inovação no setor elétrico.

Palavras-chave: Previsão de irradiação solar; Redes neurais recorrentes; LSTM; Inteligência artificial; Microgrids; Energia limpa; Geração distribuída.

Abstract: The growing technological revolution driven by artificial intelligence (AI) has profoundly impacted various sectors, including renewable energy. The pursuit of solutions to enhance the efficient production of clean energy drives the application of predictive methods based on machine learning. In this context, accurate solar irradiation forecasting is crucial for optimizing photovoltaic system performance and automating microgrid operations. This study explores the application of recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) cells to predict solar irradiation using historical data and weather conditions. The methodology presented offers a promising solution to enhance the reliability and efficiency of renewable energy integration into power systems, contributing to a more sustainable energy landscape. The results demonstrate the potential of LSTM networks in solar irradiation forecasting and highlight the relevance of AI-based solutions for the ongoing energy transition, supporting distributed generation technologies and responding to the growing demand for innovation in the electric sector.

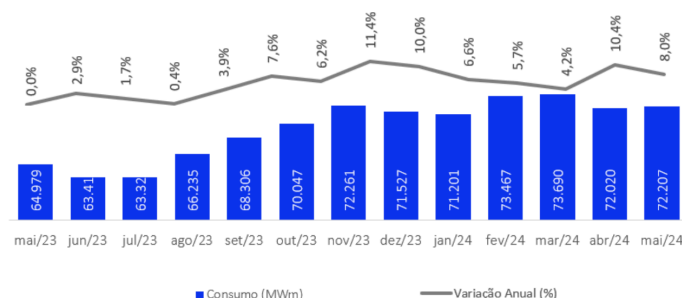
Keywords: Solar irradiation forecasting; Recurrent neural networks; LSTM; Artificial intelligence; Microgrids; Clean energy; Distributed generation.

I. INTRODUÇÃO

A previsão de irradiação solar é um elemento essencial para otimizar a geração e distribuição de energia em sistemas renováveis. A energia solar, uma fonte de energia limpa e sustentável, tem ganhado destaque diante dos desafios climáticos e da necessidade de diversificação das matrizes energéticas globais [1]. Contudo, as oscilações dinâmicas das condições climáticas tornam necessária a implementação de algoritmos preditivos eficientes para maximizar a previsão da irradiação solar e seu aproveitamento adequado [2].

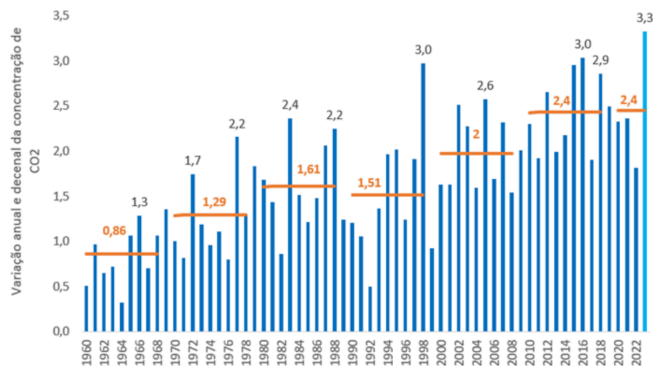
De acordo com dados recentes da Câmara de Comercialização de Energia Elétrica (CCEE), o Brasil encerrou maio de 2024 com um consumo de 70.207 megawatts médios (figura 1), o que representa um aumento de 8% em relação ao mesmo período do ano anterior. Esse aumento reflete a crescente demanda por eletricidade, exigindo soluções inovadoras para a gestão eficiente da energia elétrica. Paralelamente, estudos indicam que a taxa de aumento do dióxido de carbono atmosférico é 10 vezes mais rápida do que em qualquer outro período dos últimos 50 mil anos (figura 2), reforçando a urgência de mitigação de impactos ambientais por meio de fontes energéticas sustentáveis.

Figura 1– Evolução do Consumo De Energia em 12 Meses no Brasil



Fonte: CCEE, 2024

Figura 2 – Crescimento Anual da Concentração de CO₂ do Laboratório Mauna Loa (1960-2023)



Fonte: NOAA, Global Monitoring Laboratory

Neste cenário, o presente artigo propõe o desenvolvimento de um algoritmo baseado em redes neurais recorrentes (RNN) com unidades de memória de longo prazo (LSTM) para previsão de irradiação solar a partir de dados históricos coletados durante quatro meses [3]. As RNNs LSTM destacam-se por sua capacidade de capturar padrões temporais complexos em dados sequenciais, melhorando a precisão das previsões mesmo em cenários de elevada variabilidade climática. Trabalhos anteriores demonstraram que previsões precisas de irradiação solar impactam diretamente a eficiência de microgrids, particularmente ao integrar geração distribuída para otimizar custos energéticos e aumentar a penetração de energias renováveis no mercado [4].

Para compreender o estado da arte e contextualizar os desafios técnicos envolvidos, revisamos estudos prévios que exploram a previsão de irradiação solar e seus impactos no planejamento de sistemas energéticos. Em nossas pesquisas, foram analisados diferentes cenários de previsão utilizando dados climáticos reais combinados com o modelo de aprendizado de máquina, e a capacidade de aprendizado da LSTM foi de grande utilidade para nossos testes. Sendo obtidos ótimos resultados usando apenas uma entrada para a previsão do próximo ponto de operação, calcula a cada 10 intervalos de tempo.

Ao considerar aplicações industriais e comerciais, observa-se que a previsão de irradiação solar não apenas otimiza a operação de sistemas fotovoltaicos, mas também contribui para a estabilidade da rede elétrica em regiões com alta penetração de geração distribuída. Estudos como [5] apontam, por exemplo, a possibilidade de previsão de perdas em redes de distribuição por meio também de Redes Neurais Artificiais, a previsão de irradiação solar pode atuar como um elemento chave para reduzir as perdas técnicas nesses sistemas de distribuição quando combinada com técnicas de gerenciamento de carga e armazenamento de energia.

A proposta deste trabalho vai além de modelos existentes ao explorar abordagens inovadoras para lidar com a alta variabilidade dos dados climáticos. Para tal, utilizamos um conjunto de dados robusto, composto por medições horárias de irradiação solar, temperatura, umidade relativa, velocidade do vento entre outras, o que viabilizou

um estudo comparativo que analisa se o treinamento da rede se adapta melhor a previsões usando varias entradas, ou analisando apenas os dados passados de irradiação solar para prever o próximo valor de irradiação. Os dados foram coletados de diferentes regiões o que assegura também a capacidade de generalização do treinamento. Essas informações foram processadas e normalizadas para treinamento e validação do modelo de previsão.

Resultados preliminares indicam que o modelo LSTM alcança um erro absoluto médio de 24 unidades, o que é pouco considerando que a faixa de valores de irradiação é entre 0 e 1200; superando assim métodos convencionais como regressão linear e árvores de decisão. Esses resultados demonstram o potencial do modelo proposto para aprimorar a gestão de sistemas solares em escala nacional. Ademais, as simulações realizadas indicam que a previsão precisa pode auxiliar na integração mais eficaz da energia solar com outras fontes renováveis, criando um sistema energético mais resiliente e eficiente.

Em conclusão, este trabalho contribui significativamente para o desenvolvimento de soluções tecnológicas voltadas à energia solar, com implicações diretas para o setor elétrico e para a sustentabilidade ambiental. Estudos futuros poderiam explorar a inclusão de outras variáveis climáticas e o impacto de eventos extremos, como tempestades e ondas de calor, para refinar ainda mais os modelos preditivos propostos.

II. FUNDAMENTOS TEÓRICOS

Após estudos preliminares, relacionados ao algoritmo de treinamento da RNN, foram identificadas 2 (duas) possíveis formas de modelagem da entrada da estrutura da Rede Neural:

- Utilização de vários parâmetros de entrada, no caso dados de tempo, umidade e temperatura para prever o próximo valor de irradiação, no mesmo intervalo de tempo
- Utilização de um único parâmetro de entrada, no caso dados passados de Irradiação para prever o próximo ponto num intervalo de 10 minutos.

A seguir são apresentados os fundamentos e as características de cada uma dessas duas formas possíveis de modelagem da microgeração distribuída.

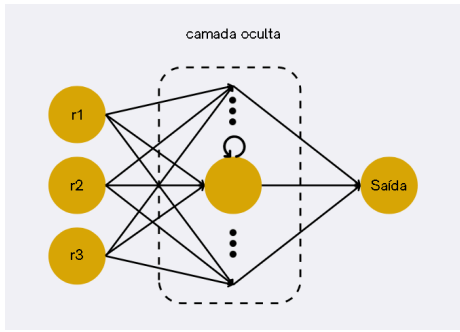
A. Multientradas

A Figura 3 apresenta um diagrama esquemático que descreve a estrutura básica de uma Rede Neural Recorrente (RNN), com a inclusão de múltiplas entradas durante o treinamento, destacando sua arquitetura projetada para lidar com dados sequenciais. Essa rede é composta por três componentes principais, cada um desempenhando um papel essencial na capacidade da RNN de capturar padrões temporais, modelar dependências entre variáveis e realizar previsões com alta precisão.

A inclusão de múltiplas entradas permite que a rede

como dados climáticos (temperatura, umidade, etc.) ou padrões cíclicos (como a periodicidade diária representada por senos e cossenos)

Figura 3 – Desenho esquemático do modelo da RNN



Fonte : O Autor

A primeira parte da rede é a camada de entrada, onde os dados são fornecidos à arquitetura. No caso apresentado, as variáveis r_1 , r_2 e r_3 representam as entradas que carregam informações críticas da sequência temporal a ser analisada. Essas variáveis podem corresponder a características distintas do conjunto de dados, como variáveis climáticas ou características temporais que influenciam a variável de saída. Cada passo temporal dessas variáveis é transmitido à camada seguinte, fornecendo os dados necessários para o processamento subsequente.

A segunda parte é a camada oculta, que é a essência da arquitetura recorrente e está representada pelo contorno pontilhado no diagrama. Essa camada é composta por neurônios que possuem uma característica única: além de se conectarem às entradas e à saída, esses neurônios também estão interligados entre si, formando conexões recorrentes. Essa estrutura permite que a rede retenha informações de passos temporais anteriores e as utilize em passos subsequentes, criando uma memória temporal que é crucial para a análise de séries temporais e outras tarefas que exigem a captura de padrões sequenciais. É na camada oculta que a RNN constrói seu entendimento contextual dos dados, utilizando essas conexões internas para modelar dependências de curto e longo prazo. Isso é particularmente importante em cenários onde o comportamento atual depende de eventos passados, como a previsão de radiação solar com base em padrões históricos de irradiância, temperatura e periodicidade diária.

Por fim, a camada de saída gera os resultados da rede com base nas informações processadas pela camada oculta. Aqui, a combinação do estado atual da sequência e do histórico de passos anteriores é integrada para produzir uma predição. Essa abordagem permite que a rede capture tanto variações instantâneas quanto tendências mais amplas nos dados temporais. A saída é especialmente relevante em tarefas como a previsão temporal, onde é essencial que a rede leve em consideração o contexto histórico para melhorar a precisão das estimativas.

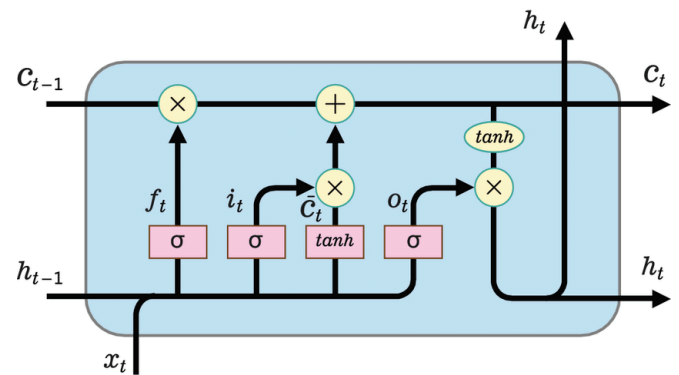
A capacidade das RNNs de reter informações de estados anteriores e utilizá-las em passos futuros torna essa arquitetura uma ferramenta poderosa para diversas aplicações. Isso inclui não apenas a análise de séries temporais, mas também problemas mais amplos, como processamento de linguagem natural, onde a dependência de contexto é crítica, e a previsão de dados climáticos, como neste caso, onde variáveis interdependentes ao longo do tempo influenciam diretamente as condições futuras. Essa

permitindo extrair padrões complexos e fornecer resultados robustos.

A.1 LSTM

Cada célula da camada oculta se trata de uma LSTM (Long Short-Term Memory), com o modelo que segue a figura 4, que é uma variação das redes neurais recorrentes projetada para lidar com dependências de longo prazo em sequências de dados. Seu principal objetivo é evitar problemas como o desaparecimento ou explosão de gradientes, o que é comum em redes recorrentes tradicionais. Para isso, ela utiliza uma estrutura organizada em portas que controlam o fluxo de informações ao longo do tempo, permitindo que a rede decida quais informações manter, descartar ou atualizar.

Figura 4– Célula LSTM padrão



Fonte : ResearchGate

Em linhas gerais a célula LSTM mantém dois tipos principais de estados: o **estado da célula** (C_t) e o **estado oculto** (h_t). O estado da célula é responsável por armazenar informações ao longo do tempo, enquanto o estado oculto serve como saída em cada etapa. Esses estados são atualizados dinamicamente por meio de três portas principais: a porta de esquecimento, a porta de entrada e a porta de saída.

Porta de Esquecimento (f_t):

A porta de esquecimento determina quais informações do estado da célula anterior (C_{t-1}) devem ser descartadas. Para isso, ela utiliza a função sigmoide, que gera valores entre 0 (descartar) e 1 (manter). Esses valores são calculados com base na combinação do estado oculto anterior (h_{t-1}) e da entrada atual (x_t):

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

Porta de Entrada (i_t) e Informação Candidata ($\bar{C}_t$):

A porta de entrada decide quais novas informações devem ser adicionadas ao estado da célula. Primeiro, a função sigmoide calcula os valores que controlam essa atualização. Em seguida, uma nova informação candidata ($\bar{C}_t$) é gerada

usando a função tangente hiperbólica (tanh), que produz valores entre -1 e 1. Essas duas etapas garantem que apenas as informações relevantes sejam incorporadas:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\bar{c}_t = \tanh(W_c * [h_{t-1}, x_t] + b_c) \quad (3)$$

Atualização do Estado da Célula (C_t)

O estado da célula atual é atualizado combinando o estado anterior com as novas informações. A porta de esquecimento (f_t) controla o quanto do estado anterior será mantido, enquanto a porta de entrada (i_t) define o quanto da nova informação candidata será incorporada:

$$C_t = f_t \odot C_{t-1} + i_t \odot \bar{c}_t \quad (4)$$

Aqui, $\odot$ representa a multiplicação elemento a elemento, garantindo que o fluxo de informações seja ajustado de forma dinâmica.

Porta de Saída (o_t)

A porta de saída decide quais informações do estado da célula atual (C_t) serão utilizadas para gerar o estado oculto (h_t). A combinação do estado oculto anterior e da entrada atual passa pela função sigmoide, que controla o quanto será "liberado". Em seguida, o estado da célula é ativado com a tangente hiperbólica e multiplicado pelos valores da porta de saída:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (5)$$

Cálculo do Estado Oculto (h_t)

O estado oculto é finalmente calculado como a multiplicação do resultado da porta de saída (o_t) pela versão ativada do estado da célula ($\tanh(C_t)$):

$$h_t = o_t \odot \tanh(C_t) \quad (6)$$

A.2 HeatMap

Um HeatMap, ou mapa de calor, é uma ferramenta de visualização de dados amplamente utilizada para representar valores de uma matriz ou tabela por meio de variações de cor. Essa técnica facilita a identificação de padrões, tendências ou correlações entre variáveis em conjuntos de dados complexos. No contexto de séries temporais e aprendizado de máquina, os HeatMaps são frequentemente empregados para explorar relações entre diferentes variáveis ou analisar o comportamento dos dados ao longo do tempo.

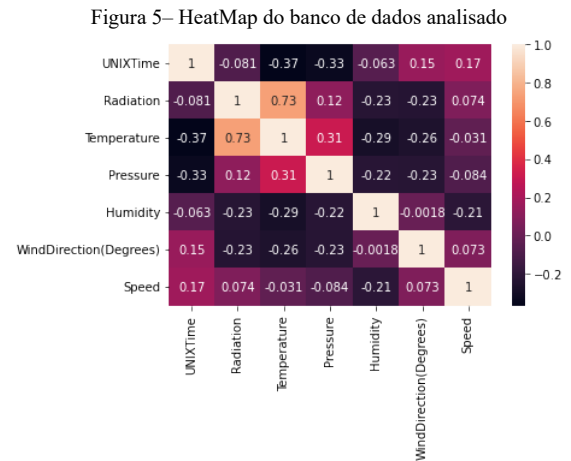
O funcionamento de um HeatMap baseia-se na atribuição de cores a valores numéricos, geralmente seguindo uma

escala contínua, como gradientes de azul para valores baixos até vermelho para valores altos. Cada célula do mapa de calor representa a interseção entre duas variáveis, permitindo uma visualização intuitiva de suas interações. Por exemplo, em uma matriz de correlação, o HeatMap é usado para exibir a força e a direção das relações lineares entre pares de variáveis.

Além disso, os HeatMaps são úteis para identificar anomalias ou padrões sazonais nos dados. Em aplicações práticas, como previsão de séries temporais, podem ser utilizados para analisar a relevância de variáveis de entrada, identificar períodos de maior concentração de valores extremos ou entender a influência de fatores externos nos dados analisados. A implementação de HeatMaps pode ser feita com ferramentas como Seaborn ou Matplotlib, que permitem personalizar escalas de cores, rótulos e anotações, melhorando a interpretação dos resultados.

Portanto, o uso de HeatMaps oferece uma abordagem visual poderosa para compreender e comunicar informações complexas, sendo uma ferramenta essencial em estudos que demandam a análise de grandes volumes de dados e suas inter-relações.

A escolha dos dados de entrada r1, r2 e r3 na arquitetura da RNN, ilustrada na figura 4, é fundamentada na análise das correlações entre as variáveis do sistema. O heatmap, apresentado na figura 5, exibe as relações de correlação entre diferentes variáveis, permitindo identificar aquelas que têm maior impacto sobre o comportamento do sistema e, portanto, são mais relevantes para o modelo.



Fonte : O Autor

As variáveis selecionadas como r1, r2 e r3 foram aquelas que apresentaram maiores coeficientes de correlação com a variável de saída “Radiation”. De modo que, propriedades como umidade e temperatura do painel fotovoltaico foram escolhidas como variáveis de entrada (r1 e r2), por terem forte influência com a irradiação. Além disso, foi incluído o tempo como uma entrada auxiliar (r3), sendo que o mesmo foi transformado em uma função trigonométrica, utilizando os segundos dos dias em que a medição foi realizada, como exposto nas equações (8) e (9), para o processo interpretá-lo como cíclico, o que permite a rede capturar tanto os padrões diretos quanto os dinâmicos.

$$Tempo_{seno} = seno(segundos * (2 * \frac{\pi}{24*3600})) \quad (7)$$

$$Tempo_{cos} = cos(segundos * (2 * \frac{\pi}{24*3600})) \quad (8)$$

Essa seleção de variáveis garante que o modelo seja alimentado com informações relevantes e interdependentes, o que potencializa sua capacidade de aprender padrões e prever resultados de forma eficiente, mesmo em condições climáticas ou operacionais variadas.

A.3 TensorFlow e Keras

O TensorFlow e o Keras são ferramentas essenciais para o desenvolvimento de modelos de aprendizado profundo, especialmente em tarefas que envolvem séries temporais, como a previsão de irradiação solar. O TensorFlow, desenvolvido pelo Google, é uma biblioteca de código aberto altamente escalável e eficiente, projetada para realizar operações matemáticas complexas e suportar o treinamento de modelos em hardware de alto desempenho, como GPUs e TPUs. O Keras, por sua vez, integrado como uma API de alto nível do TensorFlow, proporciona uma interface amigável e modular para a construção de redes neurais, permitindo que os usuários se concentrem nos aspectos conceituais e experimentais do modelo.

No modelo desenvolvido para este trabalho, diversas funcionalidades do TensorFlow e Keras foram utilizadas, destacando-se pela simplicidade e eficiência na implementação. A arquitetura foi criada utilizando a classe Sequential, que organiza as camadas de forma linear e é ideal para redes neurais com fluxo sequencial de dados, como as LSTMs empregadas. A camada LSTM foi a base do modelo, configurada para capturar padrões temporais e dependências de longo prazo nos dados de entrada. Essa camada permite que o modelo armazene e acesse informações relevantes ao longo do tempo, sendo fundamental para a análise de séries temporais.

Para potencializar o aprendizado do modelo, foi utilizada a camada Bidirecional, que duplica a direção de processamento da LSTM, permitindo que ela analise os dados tanto no sentido cronológico quanto no inverso. Essa abordagem é especialmente vantajosa em problemas onde os padrões temporais podem ser influenciados por eventos anteriores e posteriores ao ponto de interesse.

Além das camadas LSTM, o modelo incluiu camadas densas (Dense) que realizam transformações lineares e não lineares nas representações internas, conectando as saídas das LSTMs à previsão final. A camada de saída foi configurada com um único neurônio para prever o valor da irradiação solar em cada etapa temporal.

No treinamento, o modelo foi compilado com o otimizador Adam, conhecido por sua eficiência em tarefas de aprendizado profundo devido ao ajuste adaptativo da taxa de aprendizado. A função de perda escolhida foi o Mean Squared Error (MSE), apropriada para problemas de regressão, pois minimiza o erro quadrático entre os valores preditos e os reais. Para melhorar a convergência e a estabilidade do treinamento, o callback ReduceLROnPlateau foi empregado. Este recurso ajusta dinamicamente a taxa de aprendizado com base na estagnação da métrica de validação, permitindo um ajuste mais refinado nos estágios finais do treinamento.

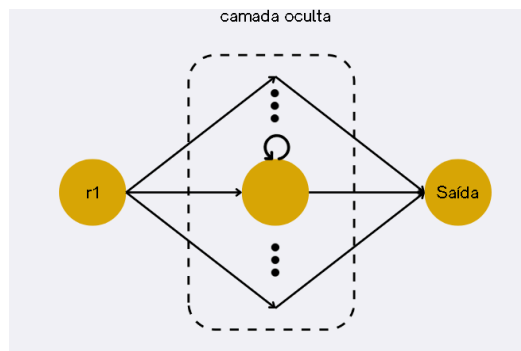
Após o treinamento, as ferramentas do Keras foram utilizadas para realizar as previsões com a função predict, integrando-se perfeitamente ao pipeline construído. Assim, o TensorFlow e o Keras se destacam não apenas pela facilidade de uso, mas

também por oferecerem soluções avançadas para problemas complexos, como o processamento de dependências temporais. Essas bibliotecas foram essenciais para garantir a eficiência, flexibilidade e rigor científico na implementação do modelo.

B Apenas uma entrada

A Figura 6 apresenta um diagrama esquemático que descreve a estrutura básica de uma Rede Neural Recorrente (RNN), com a inclusão de múltiplas entradas durante o treinamento. Assim como explicado para a RNN de múltiplas entradas, essa arquitetura também é projetada para lidar com dados sequenciais e capturar padrões temporais relevantes.

Figura 6 – Desenho esquemático do modelo da RNN



Fonte : O Autor

Na configuração com uma única entrada, como a variável, r1, apresentada no diagrama, a camada de entrada é responsável por receber uma única característica temporal de cada vez, que pode representar qualquer dado relevante, como temperatura, irradiação ou outra variável climática. Essa entrada é propagada diretamente para a camada oculta, que é o núcleo do modelo.

A camada oculta mantém sua característica central de conexões recorrentes, formando uma memória temporal. Nesse caso, ao invés de processar múltiplas entradas simultaneamente, a rede concentra seu aprendizado e modelagem na evolução de uma única sequência ao longo do tempo. Isso pode ser vantajoso em situações onde se deseja analisar o impacto isolado de uma variável específica sobre a previsão, reduzindo a complexidade e permitindo um entendimento mais detalhado do comportamento sequencial dessa única entrada.

A camada de saída, similarmente, gera a predição da rede a partir do estado processado na camada oculta. Nesse caso, a saída é diretamente influenciada pela única entrada e pela memória da rede sobre estados anteriores dessa variável. Essa abordagem pode ser particularmente útil para problemas com séries temporais simples, em que não há necessidade de incluir múltiplas variáveis no modelo, mas ainda assim é crucial capturar dependências temporais ao longo da sequência.

Assim, a RNN de entrada única, como a apresentada na figura, mantém a essência de sua arquitetura recorrente, mas com foco simplificado em uma única sequência de dados, permitindo modelar padrões temporais de forma clara e objetiva.

Em essência, os outros tópicos abordados em A.x se mantêm inalterados, as únicas mudanças que valem apenas serem comentadas é que, para a metodologia empregada aqui, não é mais necessário analisar-se o HeatMap explicado em A.2, e em A.3 a única diferença é que a camada que será usada a função “dense” é a própria camada de saída. O tópico A.1 se trata do mesmo que foi empregado aqui

III. ESTUDOS DE CASOS E RESULTADOS OBTIDOS

Uma vez estabelecido as metodologias para a implementação da previsão de irradiação solar, sua validação deve, necessariamente, se apoiar em uma correlação entre os desempenhos da representação elaborada diante de resultados de um caso base de um alimentador real. Para tanto, como mencionado anteriormente, estudos considerando-se os seguintes cenários:

- Cenário I (Múltiplas Entradas)
- Cenário II (Entrada Única)

A. Multientradas

O código foi implementado utilizando Python, para prever a radiação solar com base em variáveis meteorológicas e transformações temporais. Esta abordagem captura padrões temporais e sazonais, permitindo previsões precisas de radiação solar, essenciais para o planejamento e gerenciamento de sistemas solares. Segue abaixo o procedimento utilizado e os resultados obtidos com suas considerações:

Divisão do Conjunto de Dados

A divisão dos dados é um passo crucial para garantir que o modelo aprenda padrões confiáveis e seja capaz de generalizar para dados novos. Neste caso, foi adotada uma divisão temporal, com 80% das amostras alocadas para treinamento e 20% para teste. Essa divisão preserva a sequência temporal, evitando que informações do futuro sejam utilizadas para prever o passado, o que é fundamental em séries temporais. Além disso, a abordagem de dividir os dados temporalmente difere de divisões aleatórias comuns em outros problemas, porque a ordem temporal dos eventos contém informações que o modelo precisa aprender. Assim, essa estratégia reflete melhor a aplicação prática do modelo em ambientes reais.

Criação de Sequências Temporais

As redes LSTM são projetadas para lidar com dados sequenciais. Para esse fim, o código utiliza uma função personalizada chamada `create_sequences`, que converte as entradas (X) e as saídas (y) em blocos temporais. Essa função cria sequências de 10 passos no tempo (`time_steps = 10`), onde cada sequência contém informações históricas suficientes para prever o valor futuro. De modo que os 9 primeiros valores são alocados em um vetor de entrada, e o décimo em um vetor de saídas, para que assim a saída seja gerada a cada 10 passos temporais. Esse processo de "janela deslizante" é essencial porque as redes LSTM precisam de um histórico de dados para modelar dependências temporais. Por exemplo, para prever a radiação solar em um momento específico, o modelo analisa os 10 momentos anteriores, que é justamente o tempo definido para realizar uma previsão.

Estrutura da Rede Neural

A arquitetura da rede neural foi projetada para capturar as

características temporais e não lineares do problema. Ela é composta por camadas de LSTM simples, pois ao utilizar-se modelos com LSTMs bidirecionais o processo resultou em um *overfitting*, uma camada densa intermediária e uma camada de saída. A seguir, será detalhado os elementos dessa arquitetura:

Camadas LSTM Simples: A rede utiliza duas camadas de LSTM simples. Essas camadas processam os dados em uma única direção temporal (do passado para o futuro), permitindo que o modelo capture padrões sequenciais ao longo do tempo.

A primeira camada possui 5 unidades e inclui dropout de 3-10%, uma técnica que desativa aleatoriamente uma porcentagem de neurônios durante o treinamento para reduzir o *overfitting*. A segunda camada, com 5 unidades, também usa dropout, e com uma taxa de 10%.

Camada Densa Intermediária: Após as camadas LSTM, os dados são processados por uma camada totalmente conectada (dense) com 64 neurônios e ativação linear. Essa camada ajusta as dimensões dos dados antes de passá-los para a camada final.

Camada de Saída: A camada de saída possui apenas um neurônio, responsável por prever a radiação solar como um valor contínuo.

Vale ressaltar que os valores trabalhados durante o processo, foram os que deram melhor resultado para o banco de dados trabalhado durante o trabalho, os valores mencionados não são regras e podem ser diferentes para outros bancos de dados.

Compilação e Configuração do Modelo

Após a definição da arquitetura, o modelo foi compilado com os seguintes elementos:

Função de Perda (Loss): Foi utilizada a função de erro médio quadrático (Mean Squared Error - MSE), amplamente empregada em problemas de regressão. Ela penaliza grandes erros de forma mais severa do que pequenos erros, sendo ideal para prever valores contínuos como a radiação solar.

Otimizador Adam: O Adam (Adaptive Moment Estimation) combina as vantagens dos otimizadores RMSProp e Stochastic Gradient Descent (SGD). Ele ajusta dinamicamente a taxa de aprendizado para cada parâmetro do modelo com base em estimativas de momentos de primeira e segunda ordem (média e variância). Essa abordagem acelera a convergência, especialmente em problemas complexos como séries temporais.

Treinamento do Modelo

O treinamento foi realizado ao longo de 6 épocas, com um batch size de 32 amostras. Durante o treinamento, as seguintes estratégias foram implementadas:

Callback ReduceLROnPlateau: Este callback monitora a perda na validação e reduz dinamicamente a taxa de aprendizado pela metade quando o desempenho do modelo estagna por cinco épocas consecutivas. Isso evita que o modelo fique preso em mínimos locais e melhora a eficiência do treinamento.

Monitoramento de Perdas: As perdas no conjunto de

overfitting ou underfitting. Um gráfico das perdas foi gerado, como mostrado na figura 7, permitindo avaliar visualmente o desempenho do modelo.

Avaliação do Modelo

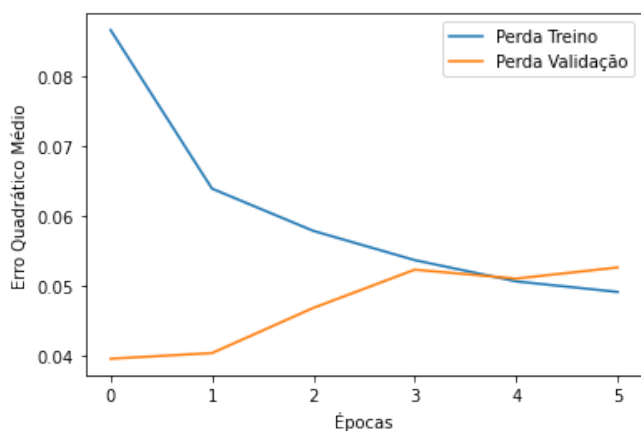
Após o treinamento, o modelo foi avaliado nos dados de teste para medir sua capacidade de generalização. As principais etapas de avaliação incluíram:

Reversão da Normalização: Tanto os valores previstos quanto os valores reais foram revertidos para sua escala original usando os objetos MinMaxScaler. Isso é necessário para calcular métricas interpretáveis no contexto original dos dados.

Erro Médio Absoluto (MAE): O MAE foi calculado para medir a precisão do modelo. Ele representa a média dos desvios absolutos entre as previsões e os valores reais, fornecendo uma métrica intuitiva e direta para avaliar o desempenho.

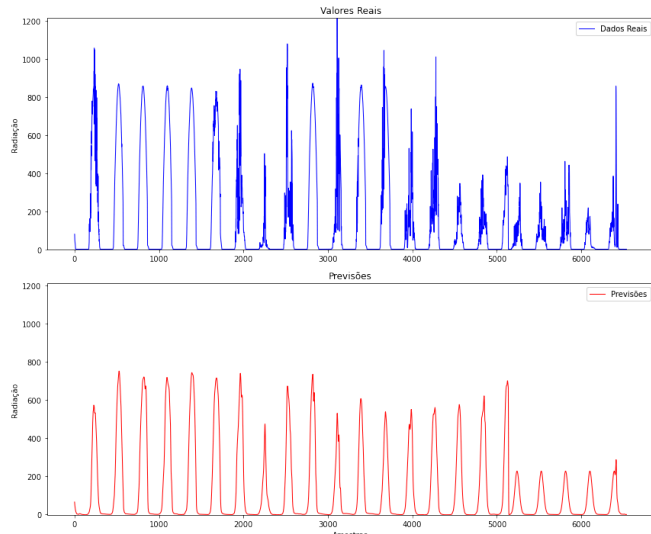
Visualização de Resultados: Um gráfico comparativo, mostrado na figura 8, entre os valores reais e as previsões foi gerado que será posteriormente comentado. Esse gráfico destaca a capacidade do modelo de acompanhar tendências e flutuações nos dados, tais resultados forneceram um erro medio absoluto igual a 63,15.

Figura 7 – Gráfico de perdas durante o processo de treinamento (multientradas)



Fonte : O Autor

Figura 8 – Comparação entre os valores reais e os valores previstos (multientradas)



B. Única entrada

O código foi implementado também utilizando Python, com os mesmos objetivos da rede de várias entradas:

Divisão do Conjunto de Dados

A divisão se manteve a mesma tomando como ideal 80% para treino e 20% para realizar as previsões

Criação de Sequências Temporais

A função create sequences também se mantém a mesma para se obter as "janelas deslizantes" com 10 momentos.

Estrutura da Rede Neural

A arquitetura da rede neural foi projetada para capturar as características temporais e não lineares do problema. Ela é composta por camadas de LSTM bidirecionais, pois renderam melhores resultados durante o processo de treinamento. A seguir, será detalhado os elementos dessa arquitetura:

Camadas LSTM Bidirecionais: A rede utiliza duas camadas de LSTM bidirecionais. Essas camadas processam os dados em ambas as direções temporais (passado para o futuro e futuro para o passado), aumentando a capacidade do modelo de capturar relações temporais complexas.

A primeira camada possui 64 unidades e inclui dropout de 10%, uma técnica que desativa aleatoriamente uma porcentagem de neurônios durante o treinamento para reduzir o overfitting. Enquanto a segunda camada possui 32 unidades sem o uso de dropout para não se perder muitos dados.

Camada de Saída: A camada de saída possui apenas um neurônio, responsável por prever a radiação solar como um valor contínuo.

Compilação e Configuração do Modelo

A função Loss e ADAM optimizer se mantém nesse modelo.

Treinamento do Modelo

O treinamento foi realizado ao longo de 15 épocas, com um batch size de 32 amostras. Durante o treinamento, as seguintes estratégias foram implementadas:

Monitoramento de Perdas: As perdas no conjunto de treinamento e validação foram monitoradas para identificar overfitting ou underfitting. Um gráfico das perdas foi gerado, como mostrado na figura 9, permitindo avaliar visualmente o desempenho do modelo.

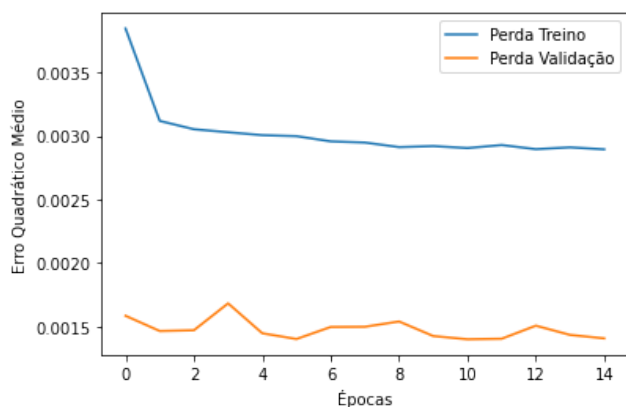
Avaliação do Modelo

As mesmas medidas de avaliação no processo de várias entradas foram aqui utilizadas

Visualização de Resultados: Um gráfico comparativo, mostrado na figura 10, entre os valores reais e as previsões foi gerado que será posteriormente comentado. Esse gráfico destaca a capacidade do modelo de acompanhar tendências

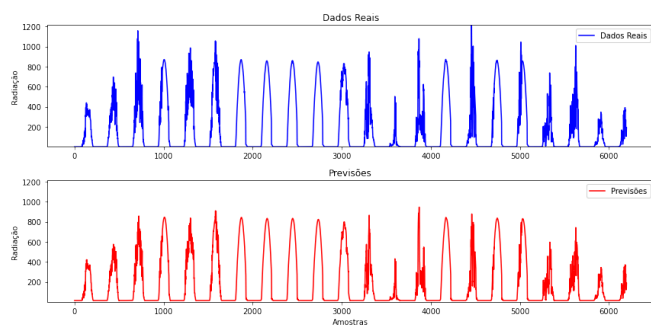
modelo anterior, tais resultados forneceram um erro medio absoluto igual a 24,06.

Figura 9 – Grafico de perdas durante o processo de treinamento (uma entrada)



Fonte : O Autor

Figura 10 – Comparação entre os valores reais e os valores previstos (unica entrada)



Fonte : O Autor

IV. CONCLUSÕES

Os resultados obtidos neste estudo demonstram a eficácia dos modelos de Redes Neurais Recorrentes (RNNs) com unidades de memória de longo prazo (LSTM) na previsão de irradiação solar. A análise comparativa entre as arquiteturas de entrada única e múltiplas entradas revelou que a primeira apresentou desempenho superior, com um erro médio absoluto (MAE) de 24,06, em relação ao MAE de 63,15 da abordagem com múltiplas entradas. Esses achados sugerem que, para o contexto analisado, a modelagem baseada exclusivamente em dados históricos de irradiação solar é suficiente para capturar padrões temporais relevantes e realizar previsões precisas.

Embora o modelo de entrada única tenha se destacado em termos de desempenho, o modelo com múltiplas entradas foi mais amplamente discutido por abordar um caso mais geral. A consideração de variáveis adicionais, como temperatura, umidade e outros dados climáticos, permite uma abordagem mais abrangente, especialmente em cenários onde múltiplos fatores influenciam a irradiação solar. Assim, ele apresenta maior potencial de adaptação a diferentes contextos e aplicações práticas, embora com maior complexidade computacional e precisão inferior neste estudo específico.

A simplicidade da abordagem de entrada única facilita sua implementação e minimiza a complexidade

situações onde dados históricos de irradiação sejam suficientes. Por outro lado, o modelo de múltiplas entradas oferece uma base promissora para estudos futuros que explorem cenários mais complexos.

Este trabalho contribui significativamente para o avanço de tecnologias voltadas à gestão de sistemas fotovoltaicos e microgrids, promovendo maior confiabilidade e eficiência na integração de energias renováveis. Estudos futuros poderão investigar a inclusão de variáveis adicionais e a aplicação de metodologias semelhantes em diferentes contextos climáticos e regionais, ampliando o potencial de generalização dos modelos.

V. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Passini, A. F. C., Niz, I. V. G., Munaretto, L. F., de Borba, W. F., & Rodrigues, A. C. (2023). Energia solar no Brasil: Oportunidades e desafios. Instituto Brasileiro de Estudos Ambientais.
- [2] Passini, A. F. C., Niz, I. V. G., Munaretto, L. F., Borba, W. F., & Rodrigues, A. C. Energia Solar no Brasil: Oportunidades e Desafios. Instituto Brasileiro de Estudos Ambientais, 2023
- [3] Dronio. Solar Energy Dataset. Kaggle, 2019. Disponível em: <https://www.kaggle.com/datasets/dronio/SolarEnergy>.
- [4] Leal, A. G. (2006). Sistema para Determinação de Perdas em Redes de Distribuição de Energia Elétrica Utilizando Curvas de Demanda Típicas de Consumidores e Redes Neurais Artificiais. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo
- [5] Correia, V. T. (2021). Sistema de Gerenciamento para microrredes Curitiba: Universidade Federal do Parana.
- [6] Cunha, M. V. da. (2016). Estratégias de gerenciamento pelo lado da demanda aplicadas aos consumidores de BT considerando a tarifa branca e a geração distribuída. Universidade Federal de Santa Maria.
- [7] Lara Filho, M. O. de. (2021). Modelo robusto orientado a dados para programação diária de operação de microrredes considerando recursos energéticos distribuídos sob incertezas.
- [8] Saúde, L. M. S. (2018). Análise comparativa entre os métodos autoregressivo, integrado de médias móveis e rede neural artificial para previsão de séries temporais.
- [9] Bissiriu, A. O. S. (2023). Contribuições para o gerenciamento de energia de microrredes CA monofásicas usadas em comunidades isoladas. Natal: Universidade Federal do Rio Grande do Norte.
- [10] Alves, F. A. Uma análise das smart grids aplicadas às microrredes inteligentes. Sobral: Universidade Federal do Ceará.
- [11] Rauber, T. W. Redes neurais artificiais. Universidade Federal do Espírito Santo.