

Previsão de irradiância solar em longo prazo com redes neurais.

Ricardo Machado

Estudante de Engenharia Elétrica

Universidade Federal da Bahia

Salvador, Brasil

machado.ricardo@ufba.br

Ricardo Pinto

Estudante de Engenharia Elétrica

Universidade Federal da Bahia

Salvador, Brasil

pinto.ricardo@ufba.br

Resumo—A previsão de irradiância solar em longo prazo é uma ferramenta importante para o estudo da implementação de uma usina fotovoltaica numa região. Considerando a necessidade de transição energética, o objetivo do trabalho é utilizar arquiteturas de redes neurais para prever a irradiância solar na cidade de Natal, no estado do Rio Grande do Norte, com base nos dados meteorológicos disponíveis pelo National Solar Radiation Database(NSRDB). Esse trabalho compara o desempenho de um Multilayer Perceptron, um Long Short Term Memory e um Gated Recurrent Unit para um conjunto de teste composto por dois anos de dados disponíveis(2023 e 2024) e um conjunto de treino composto pelos anos de 2020 a 2022. O melhor resultado obtido foi com o LSTM, que apresentou uma raiz do erro quadrático médio (RMSE) de 0,079195 e um erro absoluto médio (MAE) de 0,047601.

Index Terms—Irradiância, Redes Neurais, Energia Solar, Long Short Term Memory, Multilayer Perceptron, Gated Recurrent Unit, Usinas Fotovoltaicas

I. INTRODUÇÃO.

A irradiância solar é definida como a potência que o Sol irradia por área de superfície. A irradiância global horizontal (GHI) é composta tanto pela componente direta, que atinge a superfície da Terra diretamente, quanto pela componente difusa, que é espalhada pela atmosfera [1].

No contexto de transição energética, faz-se necessário buscar alternativas para os combustíveis fósseis, que contribuem significativamente para as mudanças climáticas e representam 80,9% da matriz energética mundial em 2022 [2]. Ao mesmo tempo, as energias renováveis como a solar e a eólica apresentam menor robustez, por dependerem de fatores climáticos variáveis. O planejamento de usinas fotovoltaicas depende de estudos sobre o potencial solar de uma determinada localidade. Para a estimação do potencial solar, *softwares* como o PVSyst utilizam dados climáticos [3], incluindo o GHI, para estimar a capacidade de geração de uma usina fotovoltaica em uma determinada localidade.

Considerando a importância da irradiância global horizontal na geração fotovoltaica. Esse trabalho tem como objetivo usar arquiteturas de redes neurais para prever a irradiância solar no período de dois anos com base em dados climáticos de uma região.

II. REVISÃO BIBLIOGRÁFICA.

O artigo de Khan et al [4] aborda a previsão de irradiância solar no período de curto prazo para a cidade de Karachi, no Paquistão, e compara o desempenho de diferentes arquiteturas de redes neurais: o Multi Layer Perceptron, a Rede Neural Recorrente, o Long Short Term Memory e o Gated Recurrent Unit. A métrica de perdas utilizada para treinamento dos modelos foi o erro absoluto médio (MAE). Já para avaliação de desempenho utiliza-se a raiz do erro quadrático médio (RMSE).

O estudo de caso [5] comenta sobre como a ausência de estações meteorológicas dificulta a aquisição de dados de irradiância solar e realiza a previsão de GHI para a cidade de Bajahol, na Índia, com base em dados de satélites que englobam um período de 11 anos.

A previsão de irradiância solar é dividida entre curtíssimo prazo, curto prazo e longo prazo. O curtíssimo prazo consiste na previsão realizada em até 3 horas adiante e requer dados climáticos em tempo real. Já a previsão em curto prazo engloba o período de alguns dias. Por fim, a previsão em longo prazo abrange períodos anuais e é utilizada para estimar a geração de energia solar no futuro e projetar novas usinas fotovoltaicas [6].

O efeito de diferentes parâmetros meteorológicos na previsão de irradiância solar é discutido em [7]. O trabalho compara o desempenho de uma Rede Neural Artificial, uma LSTM e o modelo XGBoost na previsão em curto prazo. O melhor resultado obtido para o conjunto de teste ocorreu quando os atributos de entrada eram apenas a irradiância cumulativa anterior e a precipitação total.

III. DESCRIÇÃO DOS MÉTODOS UTILIZADOS.

A. Apresentação do banco de dados utilizado.

O banco de dados utilizado no estudo é o National Solar Radiation Database(NSRDB), um *dataset* do Departamento de Energia dos Estados Unidos que engloba dados meteorológicos provenientes de satélites geoestacionários para o continente americano com alta precisão espacial e temporal.

Para o presente trabalho, foram utilizados dados meteorológicos da cidade de Natal, capital do Rio Grande do Norte, dos anos de 2020 até 2024 e com intervalo entre

amostras de 30 minutos para realizar a previsão de irradiância solar no período de longo prazo. Os atributos utilizados para treinamento dos modelos preditivos foram definidos com base na revisão sistemática [5] e são listados na tabela I. A escolha da cidade deu-se pela proximidade com a linha do Equador, o que confere à região elevados índices de irradiância solar e menor variabilidade sazonal ao longo do ano. Ademais, o estado do RN já investiu R\$ 543 milhões em energia solar até o fim do ano de 2021 [8].

Tabela I
LISTA DE ATRIBUTOS UTILIZADOS.

Nome do atributo	Classificação	Tipo
Irradiância Horizontal Global	Atributo alvo	Númerico
Temperatura Ambiente	Atributo preditivo	Númerico
Umidade Relativa do Ar	Atributo preditivo	Númerico
Pressão Atmosférica	Atributo preditivo	Númerico
Direção do vento	Atributo preditivo	Númerico
Velocidade do Vento	Atributo preditivo	Númerico
Ponto de Orvalho	Atributo preditivo	Númerico
Dia do Ano	Atributo preditivo	Periódico
Hora do Dia	Atributo preditivo	Periódico

B. Pré-processamento de dados.

A normalização dos atributos é necessária para padronizar todos os valores em uma escala comum. Esse processo evita que valores com magnitudes maiores sejam interpretados pelas redes neurais como mais relevantes do que atributos com menores valores.

Os atributos numéricos não periódicos passam pela normalização min-max, descrita em 1. O valor do atributo é representado por X , o valor máximo por X_{\max} , o valor mínimo por X_{\min} e o valor normalizado por \bar{X} . Essa normalização específica resulta num valor entre -1 e 1.

$$\bar{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

Os atributos periódicos recebem a normalização senoidal, descrita em 2, cujo processo consiste em aplicar uma função seno com período N , igual ao do atributo. Essa normalização resulta no valor entre 0 e 1.

$$\bar{X} = \sin\left(\frac{2\pi X}{N}\right) \quad (2)$$

Considerando o principal objetivo do trabalho, que é prever a irradiância solar no futuro com base em previsões climáticas, os anos de 2020 a 2022 são definidos como o conjunto de treino do problema e os anos de 2023 e 2024 são o conjunto de teste. Essa distribuição resulta em 60% do *dataset* no conjunto de treino e 40% no conjunto de teste. A separação entre teste e validação não foi realizada, pois os dois conjuntos teriam dados muito semelhantes devido à natureza temporal presente. Além disso, a criação de um conjunto de validação reduziria o intervalo de previsão para apenas um ano.

C. Redes neurais utilizadas.

1) *Multilayer Perceptron*: A estrutura de um Multilayer Perceptron (MLP) é formada por várias camadas compostas por neurônios, que são totalmente conectados entre camadas adjacentes. O modelo da figura 1 apresenta uma camada de entrada, uma camada de saída e duas camadas ocultas, as variáveis W_1 e W_2 representam os pesos sinápticos e B representa o *bias* (viés) da rede neural.

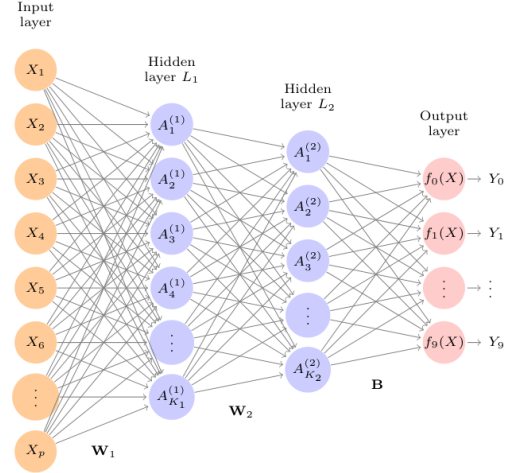


Figura 1. Estrutura de um Multilayer Perceptron. Fonte: [9]

Nas camadas ocultas, cada neurônio aplica uma função de ativação aos valores de entrada ponderados, que introduz uma não linearidade à rede neural. No caso do MLP do trabalho, é utilizada a função ReLU, descrita em 3, como função de ativação.

$$\text{ReLU}(x) = \max(0, x) = \begin{cases} x & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases} \quad (3)$$

2) *Long Short Term Memory*: O Long Short Term Memory é uma rede neural recorrente adaptada para ser capaz de lidar com dependências de longo prazo. Dessa forma, as LSTMs apresentam uma realimentação na camada oculta conforme exibido na figura 2 [10].

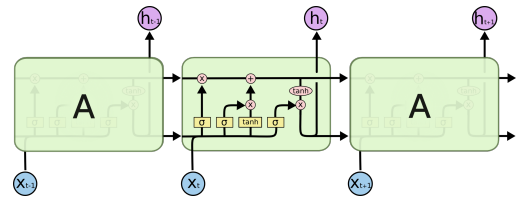


Figura 2. Estrutura de uma LSTM. Fonte: [10]

A LSTM é composta por três gates, que representam estruturas capazes de controlar a informação armazenada no estado celular. A presença do estado celular justifica a capacidade dessa rede aprender dependências de longo prazo.

O gate de esquecimento (Fig. 3) determina quais informações devem ser descartadas do estado celular anterior.

Ele utiliza uma função sigmoide que produz valores entre 0 e 1 para cada componente do estado celular, onde 0 significa "descartar completamente" e 1 significa "manter totalmente". Já o gate de entrada (Fig. 4) controla quais novas informações serão armazenadas no estado celular e atua em conjunto com uma camada tanh que cria um vetor de novos valores candidatos. Essa estrutura também decide quais desses valores candidatos serão efetivamente incorporados ao estado celular.

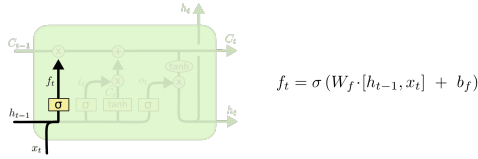


Figura 3. Gate de esquecimento da LSTM. Fonte: [10]

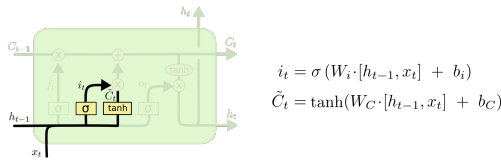


Figura 4. Gate de entrada da LSTM. Fonte: [10]

Por fim, o gate de saída (Fig. 5) regula quais partes do estado celular serão expostas como saída da célula LSTM e o estado oculto do próximo passo.

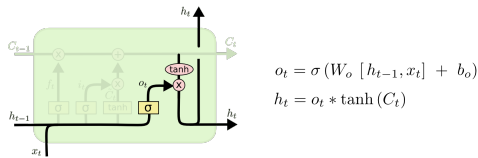


Figura 5. Gate de saída da LSTM. Fonte: [10]

3) *Gated Recurrent Unit.*: O Gated Recurrent Unit é uma rede neural recorrente com estrutura derivada da LSTM. Nessa arquitetura, o gate de entrada e esquecimento são unificados no gate de atualização. Ademais, o estado oculto e o estado celular são mesclados [10].

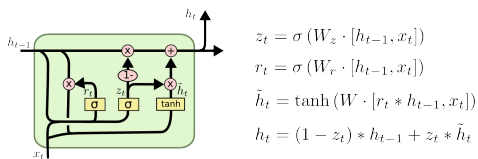


Figura 6. Estrutura de um GRU. Fonte: [10]

D. Fluxograma do projeto

A figura 7 representa a metodologia proposta para o trabalho. As etapas iniciais consistem no carregamento e no pré-processamento de dados. Em seguida, a otimização dos hiperparâmetros é realizada para os três modelos propostos

com auxílio de *framework*, que realiza esse processo até obter hiperparâmetros que sejam considerados otimizados. Nesse trabalho, utiliza-se a configuração padrão do *framework* Optuna. Com base nos modelos obtidos, realiza-se o treinamento dos modelos finais e, por fim, a avaliação das redes neurais propostas com base nas métricas de desempenho.

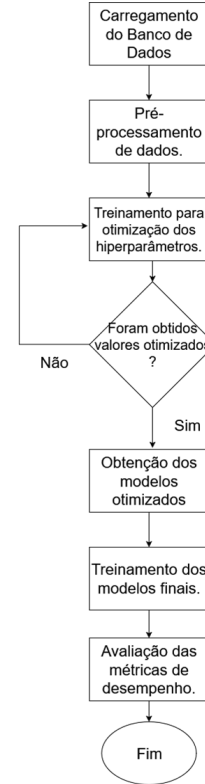


Figura 7. Fluxograma proposto para o trabalho.

E. Frameworks utilizados.

A otimização dos hiperparâmetros dos modelos é realizada utilizando o *framework* Optuna [11]. O processo de otimização consiste na amostragem aleatória para selecionar possíveis combinações de hiperparâmetros dentro de intervalos pré-definidos, o que permite uma busca eficiente no espaço de configurações possíveis. Nesse processo, delimita-se o máximo de duas camadas ocultas para os modelos e o número de neurônios por camada entre 5 e 30. Por fim, escolhe-se otimizar também a taxa de aprendizagem e o decaimento de pesos do modelo.

Os modelos MLP, LSTM e GRU foram implementados utilizando o *framework* PyTorch. Para prevenir o sobreajuste das redes neurais e garantir melhor generalização, foi aplicada a técnica de early stopping, que monitora a perda no conjunto de teste durante o treinamento e interrompe o processo quando não há melhoria por um número predefinido de épocas. Adicionalmente, foi implementado o decaimento de pesos (weight decay) como forma de regularização L2, penalizando pesos com magnitudes elevadas [7].

F. Métricas de desempenho.

Na revisão bibliográfica realizada, as principais métricas de desempenho utilizadas na literatura são o erro absoluto médio (MAE), a raiz do erro quadrático médio (RMSE) e o coeficiente R^2 [12], [4], [13]. Nesse trabalho, o desempenho é avaliado com base nas duas primeiras medidas.

1) *Erro absoluto médio*.: O erro absoluto médio é descrito por 4, em que n representa o número de amostras, y_i é o valor real do atributo alvo para a i -ésima amostra e \hat{y}_i é o valor previsto do atributo alvo para a i -ésima amostra.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4)$$

2) *Raiz do erro quadrático médio*.: A raiz do erro quadrático médio é descrita por 5. Essa medida é utilizada como a função de perda no treinamento das redes neurais.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5)$$

IV. RESULTADOS.

A. Otimização dos hiperparâmetros.

A etapa de otimização dos hiperparâmetros para os três modelos levou 5 horas e 42 minutos num processador Ryzen 5 5600x. Os valores otimizados para o MLP, o LSTM e o GRU são listados na tabela II.

Tabela II
HIPERPARÂMETROS OBTIDOS PARA AS ARQUITETURAS DE REDES NEURASIS.

Hiperparâmetro	MLP	LSTM	GRU
Camadas Ocultas	2	2	1
Neurônios por Camada	16	21	24
Taxa de Aprendizagem	$9,54 \times 10^{-5}$	$1,80 \times 10^{-3}$	$3,98 \times 10^{-5}$
Decaimento de Pesos	$4,67 \times 10^{-5}$	$1,92 \times 10^{-6}$	$3,24 \times 10^{-3}$

B. Multilayer Perceptron.

O gráfico das perdas obtidas no modelo do Multilayer Perceptron para os conjuntos de treino e teste é exibido na figura 8.

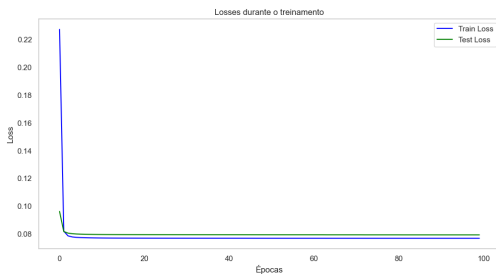


Figura 8. Gráfico de perdas para o MLP.

Para o MLP obteve-se uma raiz do erro quadrático médio de 0,079322 e um erro absoluto médio de 0,048410.

C. Long Short Term Memory.

O gráfico das perdas obtidas no modelo do Long Short Term Memory para os conjuntos de treino e teste é exibido na figura 9. O *early stopping* é ativado após 63 épocas de treinamento. Pela escala do eixo vertical, o início da curva de teste foi truncado da figura.

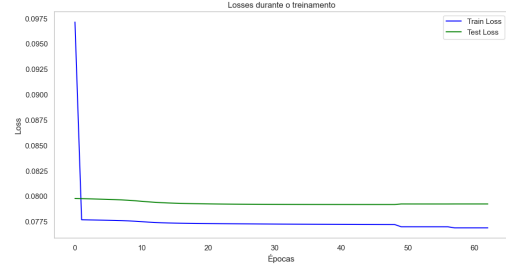


Figura 9. Gráfico de perdas para o LSTM.

Para o LSTM obteve-se uma raiz do erro quadrático médio de 0,079195 e um erro absoluto médio de 0,047601.

D. Gated Recurrent Unit.

O gráfico das perdas obtidas no modelo do Gated Recurrent Unit para os conjuntos de treino e teste é exibido na figura 10. O *early stopping* é ativado após 67 épocas de treinamento.

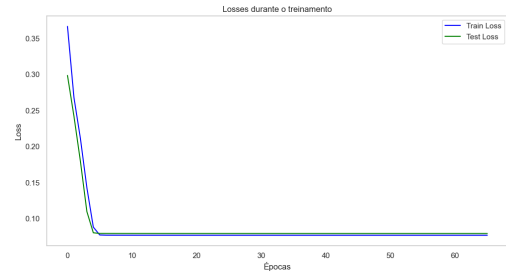


Figura 10. Gráfico de perdas para o GRU.

Para o GRU obteve-se uma raiz do erro quadrático médio de 0,079328 e um erro absoluto médio de 0,048785.

E. Tempo de execução dos modelos.

Conforme a figura 11, o MLP apresentou o pior tempo de execução: 691,63 segundos. O GRU apresentou o tempo de execução mais rápido: 257,88 segundos, esse comportamento é esperado pois essa rede neural apresenta apenas uma camada oculta ao invés das duas camadas ocultas presentes nos outros modelos. O tempo de 305,54 segundos do LSTM é um pouco maior do que o tempo de execução do GRU.

V. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.

Os valores de erro absoluto médio e raiz do erro quadrático médio obtidos indicam que as redes neurais treinadas apresentam desempenho satisfatório em prever a irradiância solar no intervalo de dois anos. O melhor desempenho na previsão ocorre com o *Long Short Term Memory*, justamente

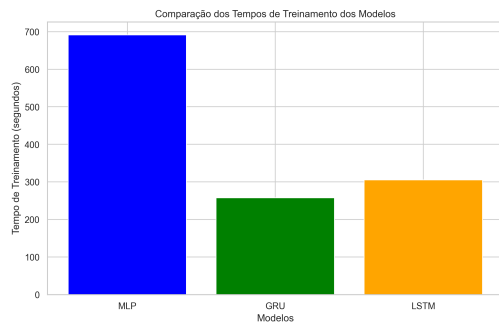


Figura 11. Comparação entre o tempo de treinamento dos modelos.

porque essa rede neural recorrente é projetada para aprender dependências de longo prazo. Já o tempo de treinamento mais curto foi obtido com o *Gated Recurrent Unit*.

Como sugestões para trabalhos futuros, é recomendado trabalhar com LSTMs mais profundas e com maior número de neurônios por camada. Outra sugestão consiste em utilizar um conjunto de dados com menor intervalo entre as amostras ou adicionar mais anos no *dataset*.

REFERÊNCIAS

- [1] Global Solar Atlas, “Frequently asked questions.” <https://globalsolaratlas.info/support/faq>, 2025. Acesso em: 10 jul. 2025.
- [2] Empresa de Pesquisa Energética, “Matriz energética e elétrica.” <https://www.epe.gov.br/pt/abcdenergia/matriz-energetica-e-eletrica>, 2024. Acesso em: 20 jul. 2025.
- [3] C. Kandasamy, P. Prabu, and K. Niruba, “Solar potential assessment using pvsyst software,” in *2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)*, pp. 667–672, 2013.
- [4] H. A. Khan, M. T. Khan, S. A. Alqahtani, A. Rehman, M. Ahmad, S. M. Altowaijri, and A. A. Khan, “Solar irradiance forecasting using deep learning techniques,” *Engineering Proceedings*, vol. 46, no. 1, p. 15, 2023.
- [5] S. Tajjour, S. S. Chandel, M. A. Alotaibi, H. Malik, F. P. García Márquez, and A. Afthanorhan, “Short-term solar irradiance forecasting using deep learning techniques: A comprehensive case study,” *IEEE Access*, vol. 11, pp. 119851–119861, 2023.
- [6] S. M. Awan, Z. A. Khan, and M. Aslam, “Solar generation forecasting by recurrent neural networks optimized by levenberg-marquardt algorithm,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 276–281, 2018.
- [7] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*, ch. Weight Decay. Cambridge University Press, 2023. Versão online em português.
- [8] W. Galvão, “Sol, habitante do rn,” 2021. Publicado pela UFRN, Natal. Acesso em 12 de julho de 2025.
- [9] G. James, D. Witten, T. Hastie, R. Tibshirani, P. Bühlmann, and F. Wilms, *Introduction to Statistical Learning with Python*. Springer Texts in Statistics, Springer, 2023.
- [10] C. Olah, “Understanding lstm networks,” August 2015. Blog post.
- [11] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” *CoRR*, vol. abs/1907.10902, 2019.
- [12] K. Olcay, S. Gıray Tunca, and M. Arif Özgür, “Forecasting and performance analysis of energy production in solar power plants using long short-term memory (lstm) and random forest models,” *IEEE Access*, vol. 12, pp. 103299–103312, 2024.
- [13] Y.-J. Zhong and Y.-K. Wu, “Short-term solar power forecasts considering various weather variables,” in *2020 International Symposium on Computer, Consumer and Control (IS3C)*, pp. 432–435, 2020.