UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Jose Alberto Cumbicos Romero

Hourly GHI Data Estimation from Daily Measurements Using Machine Learning Techniques.

Estimativa de dados GHI horários a partir de medições diárias usando técnicas de aprendizado de máquina.

Campinas
2024

Jose Alberto Cumbicos Romero

Hourly GHI Data Estimation from Daily Measurements Using Machine Learning Techniques.

Estimativa de dados GHI horários a partir de medições diárias usando técnicas de aprendizado de máquina.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica, na área de Telecomunicações e Telemática.

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Telecommunications and Telematics.

Supervisor/Orientador: Prof. Dr. Gustavo Fraidenraich

Este trabalho corresponde à versão final da Dissertação defendida por Jose Alberto Cumbicos Romero e orientado pelo Prof. Dr. Gustavo Fraidenraich.

Campinas
2024

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

C91h

Cumbicos Romero, Jose Alberto, 1995-
Hourly GHI data estimation from daily measurements using machine learning techniques / Jose Alberto Cumbicos Romero. – Campinas, SP : [s.n.], 2024.

Orientador: Gustavo Fraidenraich.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação.

1. Aprendizado de máquina. 2. Redes neurais (Computação). 3. Irradiação solar. I. Fraidenraich, Gustavo, 1975-. II. Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações Complementares

Título em outro idioma: Estimativa de dados GHI horários a partir de medições diárias usando técnicas de aprendizado de máquina
Palavras-chave em inglês:
Machine learning
Neural networks (computer)
Solar irradiation
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Gustavo Fraidenraich [Orientador]
Diana Cristina González González
Denis Gustavo Fantinato
Data de defesa: 01-08-2024
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-6817-5566
- Currículo Lattes do autor: http://lattes.cnpq.br/4512147286188546

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Jose Alberto Cumbicos Romero. **RA:** 218515

Data da Defesa: 1 de agosto de 2024.

Título da Tese: "Hourly GHI Data Estimation from Daily Measurements Using Machine Learning Techniques."

Prof. Dr. Gustavo Fraidenraich (Presidente) Universidade Estadual de Campinas.

Profa. Dra. Diana Cristina González González Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis Gustavo Fantinato Universidade Estadual de Campinas

Ata da defesa, assinada pelos membros da Comissão Examinadora, encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-graduação da Faculdade de Engenharia Elétrica e de Computação.

Todo lo que estas viviendo te está preparando para todo lo que tanto has pedido, tu sigue adelante, respira, agradece y confía

(Salo)

Acknowledgements

Firstly, to God.

To Prof. Gustavo, for granting me the opportunity to become a part of his research group. I am grateful for his patience, expertise, and support have been invaluable to the success of this study.

To my family and the person who provided encouragement from afar. Your belief in me has been a motivator achieve this goal. Thank you for always being there to cheer me on and for your support.

To the *repi Humita*, my friends and colleagues, for the academic support and camaraderie. Your friendship, guidance, and shared knowledge have been instrumental to the completion of this dissertation. Thank you all for being an integral part of my academic journey.

This study was financed by Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) under Grant 88887.662210/2022-00.

I would like to thank TotalEnergies for the financial support. In addition, we are grateful to all collaborators from University of Campinas (UNICAMP). We acknowledge the support of ANP (National Agency for Petroleum, Natural Gas and Biofuels) through the R&D levy regulation. Acknowledgements are extended to the Center for Energy and Petroleum Studies (CEPETRO) and the School of Electrical and Computer Engineering (FEEC).

Resumo

Este estudo explora a aplicação de arquiteturas de aprendizado de máquina (ML) baseadas em memória, como Long Short-Term Memory (LSTM) e Gated Recurrent Units (GRU), para estimar dados horários de Irradiação Horizontal Global (GHI). Esses modelos de ML são comparados com modelos físicos tradicionais propostos por Collares-Pereira, Garg e Yao. Além disso, o Perceptron Multicamadas (MLP) e as Redes Neurais Convolucionais (CNN) são considerados e avaliados. Os modelos de ML são treinados usando uma abordagem de janela deslizante, com variáveis de entrada como irradiância diária total, ângulo horário do nascer do sol e ângulo horário solar. A otimização dos hiperparâmetros é realizada usando uma técnica de busca aleatória para melhorar o desempenho do modelo. O estudo também aborda o impacto do tamanho da janela e diferentes combinações de características de entrada no desempenho do modelo. Essa abordagem orientada por dados reduz a dependência de dados meteorológicos extensos, tornando-a aplicável em locais diversos. Os resultados indicam que os modelos de ML superam os modelos físicos, com a LSTM demonstrando o melhor desempenho geral. O desempenho superior dos modelos de ML é atribuído à sua capacidade de capturar relações complexas e não lineares nos dados. Este estudo destaca o potencial dos modelos orientados por dados na estimativa de energia solar, oferecendo uma alternativa flexível e robusta aos modelos físicos tradicionais.

Palavras-chaves: Downscaling, irradiância horizontal global, aprendizado de máquina, dados solares sintéticos.

Abstract

This study explores the application of memory-based machine learning (ML) architectures, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), for estimating hourly Global Horizontal Irradiation (GHI) data. These ML models are compared against traditional physical models proposed by Collares-Pereira, Garg, and Yao. Additionally, the Multilyer Perceptron (MLP) and the Convolutional Neural Networks (CNN) are considered and evaluated. The ML models are trained using a rolling-window approach, with input features such as total daily irradiance, sunrise hour angle, and solar hour angle. Hyperparameter optimization is performed using the random search technique to enhance model performance. The study also investigates the impact of window size and different combinations of input features on model performance. This data-driven approach reduces the reliance on extensive meteorological data, making it applicable across diverse locations. Results indicate that ML models outperform physical models, with LSTM demonstrating the best performance overall. The superior performance of ML models is attributed to their ability to capture complex, non-linear relationships in the data. This study underscores the potential of data-driven models in solar energy estimation, offering a flexible and robust alternative to traditional physical models.

Keywords: Downscaling, global horizontal irradiance, machine learning, synthetic solar data.

List of Figures

List of Tables

List of Symbols

n	Batch size
H_T	Daily irradiance
H_0	Extraterrestrial irradiance
ω	Hour angle
L	Latitud
ST	Local solar time
β_N	Solar altitude
I_{SC}	Solar constant
δ	Solar declination
ω_s	Sunrise hour angle
n	Windows size
Θ_z	Zenith Angle

List of Acronyms

CNN	Convolutional Neural Networks
DHI	Diffuse Horizontal Irradiance
DNI	Direct Normal Irradiance
GHI	Global Horizontal Irradiance
GRU	Gated Recurrent Unit
INMET	*Instituto Nacional de Meteorologia*
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine learning
MLP	Multilayer Perceptron
nRMSE	normalized RMSE
R2	R-squared
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
SONDA	*Sistema de Organização Nacional de Dados Ambientais*

List of Publications

Journal Article

- J. A. Cumbicos, L. P. Jimenéz Jimenéz, G. Fraidenraich, and T. Barros. "Hourly GHI Data Estimation from Daily Measurements Using Machine Learning Techniques." In: *IEEE Open Access Journal of Power and Energy* (2024). [Under review]

Co-Authored Conference Article

- J. C. Cortez, J. A. Cumbicos, L. Z. Terada, J. C. López, M. Giesbrecht, G. Fraidenraich, and M. J. Rider. "Fuzzy Ensemble Algorithm for Day-ahead Photovoltaic Power Forecasting." In: *2024 International Conference on Smart Energy Systems and Technologies (SEST).* [Accepted for publication]

Contents

Chapter 1

Introduction

1.1 Introduction

Solar energy plays a pivotal role in the transition to sustainable energy systems, offering a clean and renewable alternative to fossil fuel-based power generation. The efficient design, operation, and integration of solar energy systems require accurate predictions of solar irradiance, particularly Global Horizontal Irradiance (GHI). While many solar monitoring stations provide hourly resolution GHI data, there are still regions and applications where only daily data is available. In these cases, the use of high-resolution data is crucial for capturing the diurnal variations in solar radiation and improving the accuracy of solar energy system simulations [23].

In photovoltaic systems, solar data is a critical resource in several key areas. It is essential for power generation forecasting, which involves predicting the energy produced by solar panels. Additionally, solar data are used for irradiance prediction to optimize system performance. The efficiency of solar panels is also monitored using this data, helping to identify issues and inefficiencies, and making necessary adjustments to improve overall efficiency. However, a significant challenge is the lack of availability of data in the desired time resolution needed for accurate simulations.

In Brazil, there are a few accessible solar data networks that provide this information. Two notable networks are *Instituto Nacional de Meteorologia* (INMET) and *Sistema de Organização Nacional de Dados Ambientais* (SONDA). The INMET network consists of 564 automatic weather stations, with 505 currently active and 59 inactive. Each station is equipped with a pyranometer to measure GHI and collects various meteorological data such

as temperature, humidity, atmospheric pressure, precipitation, wind direction, and wind speed on an hourly basis. All the collected data are freely available and accessible on the internet [25]. Conversely, the SONDA network includes 20 stations strategically positioned to cover Brazil's diverse climate zones. These stations collect solar radiation data every minute, measuring and recording three types of solar radiation: GHI, Direct Normal Irradiance (DNI), and Diffuse Horizontal Irradiance (DHI) [39]. Fig. 1.1 illustrates the distribution of these weather stations across Brazil.

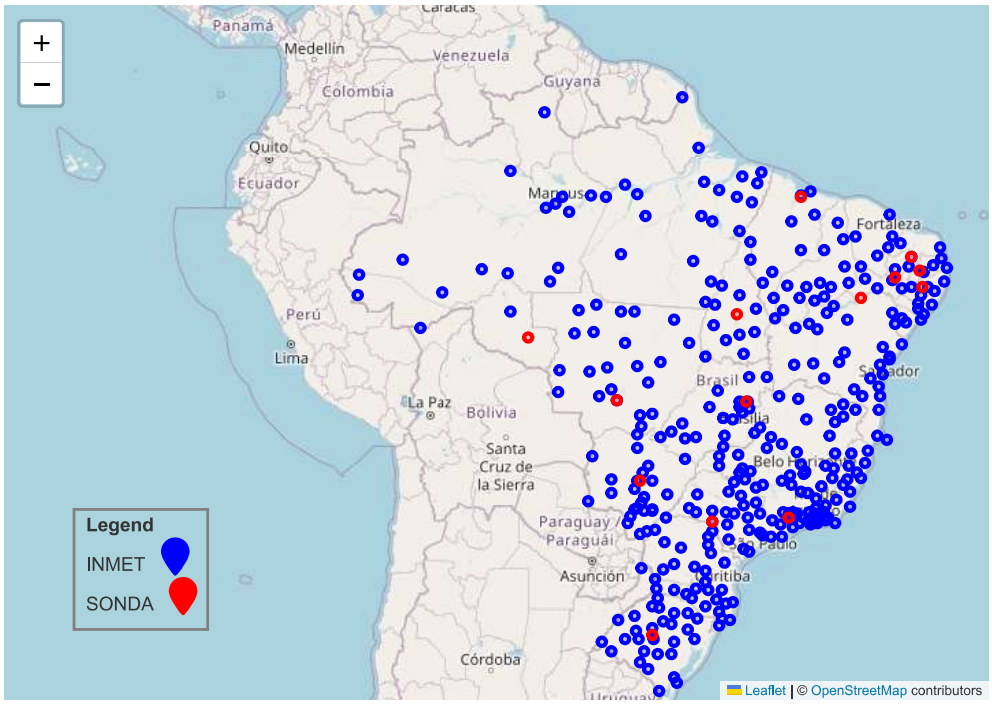


Figure 1.1: Geographical distribution of INMET and SONDA weather stations across Brazil.

The collection of solar radiation data in Brazil encounters several challenges that impact its accuracy and reliability. Despite the extensive network of weather stations, many are situated in remote areas. This geographical isolation often results in frequent equipment failure and data loss, as maintenance and repairs are difficult to perform promptly. Consequently, the data collected can be inconsistent and incomplete, making it challenging to obtain solar radiation pattern across the consecutive time periods. These issues highlight the need of implement preprocessing techniques to clean and refine the data.

In Fig. 1.1, it can be observed that there are still large areas without stations, leading to significant gaps in data collection. This issue impacts the design of photovoltaic systems, particularly in regions with complex geographical features, where higher spatial and temporal resolution data is necessary to accurately capture variations in solar radiation.

However, increasing the number of weather stations is not a feasible solution due to the high costs associated with installing and maintaining this equipment. Consequently, measured radiation data remains scarce for many locations in Brazil and worldwide. To address this gap, generating synthetic solar data can provide valuable information for regions where direct measurements are sparse or non-existent due to the lack of weather stations.

High-resolution solar irradiance data is essential for optimizing the performance of solar energy systems and maximizing their energy yield. Hourly resolution data allows for a more detailed analysis of solar radiation patterns throughout the day, capturing variations due to factors such as cloud cover, atmospheric conditions, and shading. This level of detail is particularly important for applications such as grid integration planning, where precise predictions of solar energy generation are necessary to maintain grid stability and optimize the use of solar resources.

In regions where only daily GHI data is available, the challenge lies in extrapolating this data to hourly resolution. Different works have been presented in the literature to address this problem [45, 9, 16, 47, 5, 33, 1, 28, 20]. Among these works, three distinct types of models can be identified: physical models, stochastic models, and data-driven models. Physical models, such as [9, 16, 47], consider solar geometry and atmospheric conditions to estimate the hourly variation of solar radiation. One common approach is the use of clear sky models, which calculate the solar radiation that would be received on a cloudless day and then adjust this value based on cloud cover and other atmospheric conditions. Other physical models may incorporate more complex atmospheric physics and radiation transfer equations to simulate the interaction of solar radiation with the Earth's atmosphere and surface. While physical models can provide valuable insights into the underlying processes governing solar radiation, they often require detailed knowledge of the site-specific meteorological conditions and could be computationally intensive.

Stochastic models for data downscaling utilize probabilistic approaches to simulate the variability of solar radiation at hourly intervals [33, 18, 5]. These models often incorporate stochastic processes such as Markov chains to model the transitions between different states of solar radiation, taking into account factors such as cloud cover and atmospheric conditions. Markov chains are particularly useful in stochastic modeling due to their ability to capture the memoryless property, where the probability of transitioning to a future state depends only on the current state and not on the sequence of events leading up to it. By applying Markov chains,

stochastic models can simulate the hourly and sub-hourly variability of solar radiation based on the statistical properties of the available daily data, providing insights into the stochastic nature of solar radiation patterns [1, 34]. The complexity of these models lies in establishing the transition matrix, which defines the probabilities of transitioning between different states. This matrix can vary significantly depending on the location and local climatic conditions, making it difficult to adapt the model to different places.

Regarding data-driven models, they leverage Machine learning (ML) techniques to learn the relationships between daily and hourly radiation variables, without explicitly using meteorological data in the training process. They can capture the complex and non-linear relationships that govern solar radiation patterns by focusing solely on radiation variables such as GHI, sunrise and solar hour angle. This approach eliminates the need for meteorological data, which may be unavailable or unreliable in certain locations. For instance, authors in [28] present a Generalized Regression Artificial Neural Network using the mean daily GHI, hour angle, and sunset hour angle as input parameters to estimate the mean hourly GHI. Four neurons where considered for the hidden layer obtaining an Root Mean Square Error (RMSE) of 15.1% in the test set. More recently, authors in [20] introduce another data-driven model. With the aim to capture the non-symmetric profiles of hourly GHI, authors first employed the well-known K-mean technique to group hourly observations, followed by a non-parametric function approximation using the Multilayer Perceptron (MLP) artificial neural network. The proposed model was evaluated considering a different number of hidden neurons and then compared with 15 physical models. The results showed that the ML based model outperformed, in terms of RMSE, all the physical models.

Despite the utility and accuracy of data-driven models, there is a scarcity of research employing ML techniques for this particular problem. This study seeks to address this gap by introducing Recurrent Neural Networks (RNN)s, including the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), as well as 1D Convolutional Neural Networks (CNN)s to the task of downscaling data. Specifically, the objective is to estimate hourly GHI data using daily GHI measurements, in conjunction with the sunrise hour angle and solar hour angle, as input parameters for the ML networks. The study includes a comprehensive performance evaluation, comparing these ML models with the MLP model and the physical models proposed by Garg [16], Collares-Pereira [9], and Yao [47]. Evaluation metrics such as RMSE, normalized RMSE (nRMSE), Mean Absolute Error (MAE), and R-squared (R2) are

employed. Finally, meticulous parameter optimization was conducted to achieve optimal performance of the ML models.

1.2 Objectives

The general objective of this study is to evaluate memory-based machine learning models to estimate hourly global horizontal irradiation (GHI) from daily measurements. This assessment contributes to solar energy estimation by offering a flexible and robust alternative to traditional physical models, improving the accuracy and efficiency of estimations.

1.3 Specific objectives

- To implement and optimize memory-based ML architectures, including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), and additional models such as Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN), for estimating hourly GHI data.
- To compare the performance of ML models with traditional physical models proposed by Collares-Pereira, Garg, and Yao in terms of accuracy.
- To investigate the impact of window size and different combinations of input features on the performance of the machine learning models.

1.4 Dissertation Structure

The rest of this study is organized as follows: Chapter 2 presents the fundamentals of solar radiation, including definitions and equations for calculating solar position. Additionally, it provides a brief overview of ML models considered in this study; Chapter 3 explains the methodology, describes the dataset characteristics, details the data preprocessing steps, and outlines the architecture of the models; Chapter 4 unveils and discusses the main results; Chapter 5 concludes the work and suggests directions for future research.

Chapter 2

Solar radiation and Models

This chapter offers a review of topics necessary for understanding the concepts and methodologies employed in this work. Initially, fundamental aspects of solar geometry are explained, including key concepts that describes the sun's position at any moment of the day. Subsequently, solar radiation is introduced, covering concepts such as the solar constant and extraterrestrial radiation. In the latter part of the chapter, various physical models used to estimate solar radiation are explored. Finally, machine learning models employed in this study for the estimation of GHI are introduced.

2.1 Solar geometry

Solar geometry refers to the study of the sun's position and its movement relative to the Earth. Knowing the sun's position in the sky at any time of day for any location on Earth determines the intensity that reaches the Earth's surface. This information can be used to calculate other parameters such as the optimal orientation and tilt of solar panels. Consequently, solar geometry is crucial for designing solar systems that maximize the amount of solar energy captured and converted into electricity [43].

Many algorithms have been developed to calculate the position of the sun using different techniques. For example, the Spencer method uses the Fourier series [40], while the Solar Position Algorithm developed by the National Renewable Energy Laboratory achieves greater accuracy with more complex calculations [37]. These methods are integrated into the *pvlib* Python library, which simplifies their use. Additionally, *pvlib* provides a wide range of tools and functions to handle various aspects of solar energy system analysis [24].

Solar radiation levels at different times and locations have been highly studied and are a function of angles and equations. In the following subsections, we delve into key sun position parameters, including hour angle, solar declination, zenith angle, solar altitude angle, sunrise and sunset.

2.1.1 Hour angle

The hour angle (ω) measures the time that has passed since solar noon and describes the course of the sun during a day of 24 h, expressed in angular measurement (degrees). At solar noon, the ω is zero degrees. Essentially, ω is an angular representation of solar time (ST), with each hour corresponding to fifteen degrees [41]:

$$\omega = (ST - 12) \times 15 \tag{2.1}$$

2.1.2 Solar declination

Solar declination (δ) is the angle between the plane of the equator and a line drawn from the center of the sun to the center of the Earth [17]. Fig. 2.1 illustrates how this angle varies throughout the year due to the Earth's axial tilt of approximately 23.5 degrees. In this illustration, the Earth is depicted as fixed while the sun appears to move up and down. The solar declination (δ), in degrees, for any day of the year can be calculated by sinusoidal relationship:

$$\delta = 23.45 \ \sin\left[\frac{360}{365}(n - 81)\right], \tag{2.2}$$

where n is the day number of the year. As shown in Fig. 2.1, the Tropic of Cancer, located at approximately 23.5 degrees north latitude is the farthest point north where the sun is directly overhead at noon during the summer solstice, typically on June 21. On the other hand, the Tropic of Capricorn is situated at around 23.5 degrees south latitude where the sun is overhead during winter solstice on December 21.

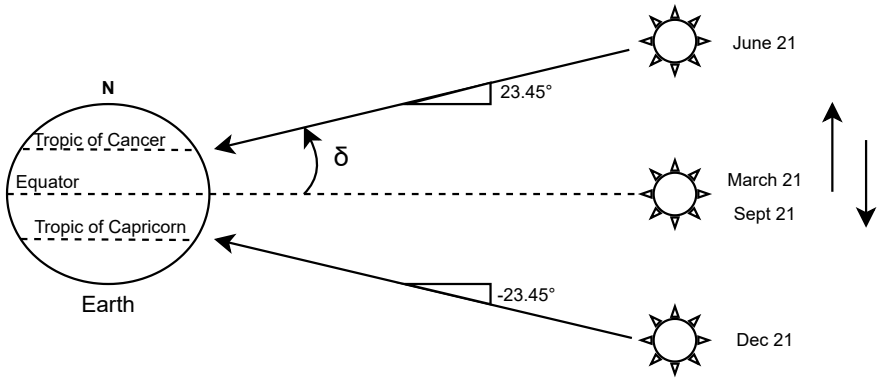


Figure 2.1: Variation of the solar declination angle with fixed earth and sun moving up and down. Adapted from [17].

2.1.3 Zenith Angle

The zenith (Θ_z) is defined as the angle between the sun's rays and an imaginary axis that is perpendicular to the local horizontal plane at the observer's location, as illustrated in Fig. 2.2. It can be observed that this angle is complementary to the solar altitude angle [17].

2.1.4 Solar altitude

Solar altitude (β_N) is the angle between the horizontal plane and the sun in the sky. Also, it can be interpreted as the sun's height in the sky and measured in degrees. Fig. 2.2 illustrates the solar altitude angle and the local horizontal plane. The mathematical expression for the solar altitude and zenith angle for any date and time is given by 2.3 [30].

$$\sin(\beta_N) = \cos(\Theta_z) = \sin(L)\sin(\delta) + \cos(L)\cos(\delta)\cos(\omega), \tag{2.3}$$

where L is the local latitude.

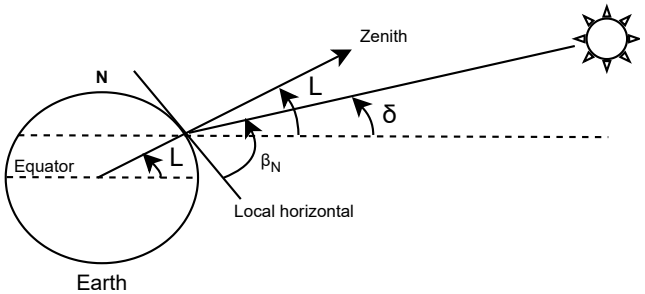


Figure 2.2: The altitude angle of the sun. Adapted from [17].

2.1.5 Sunrise and sunset

Sunrise occurs when the sun appears above the horizon in the morning. The sky gradually transitions from darkness to light, signifying the start of the day. In contrast, sunset is the moment when the sun gradually sinks below the horizon in the evening. Like sunrise, the time of sunset depends on the specific location and varies daily. At sunrise, the sun is below the horizon, this means that the angle of solar altitude is 0 degrees. Consequently, the hour angle in 2.3 represents the sunset hour angle (ω_s) when $\beta_N = 0$, and the equation can be rewritten as [35]:

$$\sin(0) = \sin(L)\sin(\delta) + \cos(L)\cos(\delta)\cos(\omega_s)$$
$$\cos(\omega_s) = -\frac{\sin(L)\sin(\delta)}{\cos(L)\cos(\delta)},$$

which reduces to:

$$\cos(\omega_s) = -\tan(L)\tan(\delta), \tag{2.4}$$

where ω_s is a positive value at sunset.

2.2 Solar radiation

Solar radiation is crucial for various applications, especially in solar power systems. This renewable and sustainable energy source reduces our dependence on fossil fuels. Solar radiation refers to the energy coming from the Sun in wavelength, forming a spectrum of electromagnetic waves [26].

The solar radiation arriving outside the Earth's atmosphere differs from that at the surface. On the Earth's surface, the amount of solar energy varies by location due to two main factors. Firstly, solar geometry, as explained in the previous section, affects the angle and position of the sun relative to a location. Secondly, atmospheric effects such as absorption and scattering attenuate solar radiation. This attenuation depends on local climatic conditions, including cloud cover, air quality, and humidity [27].

2.2.1 Solar constant

The solar constant (I_{SC}) represents the intensity of solar radiation outside the atmosphere of the Earth on a horizontal surface, as illustrated in Fig. 2.3. Using the average distance between the Earth and the Sun, the most recent estimate of the solar constant is 1361.1 W/m^2 [19]. This value can vary slightly throughout the year due to the elliptical orbit of Earth, causing changes in the distance between the Earth and the Sun [27].

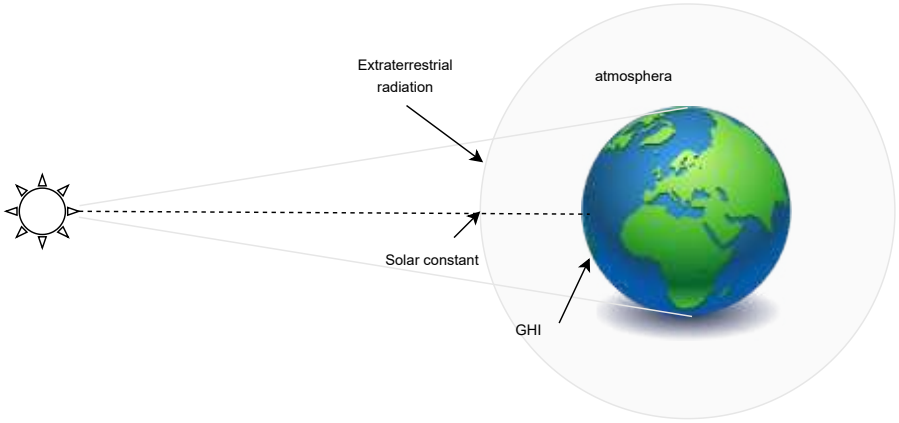


Figure 2.3: The solar constant on the top of the atmosphere. Adapted from [13].

2.2.2 Extraterrestrial Radiation

Extraterrestrial radiation (H_0) represents the maximum possible solar radiation received at the top of Earth's atmosphere if there were no atmospheric effects. Unlike I_{SC}, which provides an average value, H_0 accounts for Earth's elliptical orbit around the Sun. Then a correction factor is applied to I_{SC} to calculate H_0 for different time periods [32]. The equation for calculating extraterrestrial radiation is given by

$$H_0 = \frac{24}{\pi} I_{sc} \left[(\omega_s \sin L \sin \delta) + (\cos L \cos \delta \sin \omega_s) \right] \tag{2.5}$$

2.2.3 Solar radiation components

When solar radiation enters the Earth's atmosphere, it undergoes various processes. Some of the incident energy is absorbed by atmospheric constituents and particles, reducing the amount that reaches the ground. The radiation that arrives at the Earth's surface directly from the Sun is known as Direct Normal Irradiance (DNI). This component includes only the

direct sunlight. In contrast, scattering occurs when solar radiation is redirected in multiple directions, with some of it returning to space. This process results in Diffuse Horizontal Irradiance (DHI), which reaches the Earth's surface from all directions [38]. DHI is crucial for solar photovoltaic systems, especially on cloudy days or in shaded areas where direct sunlight is obstructed.

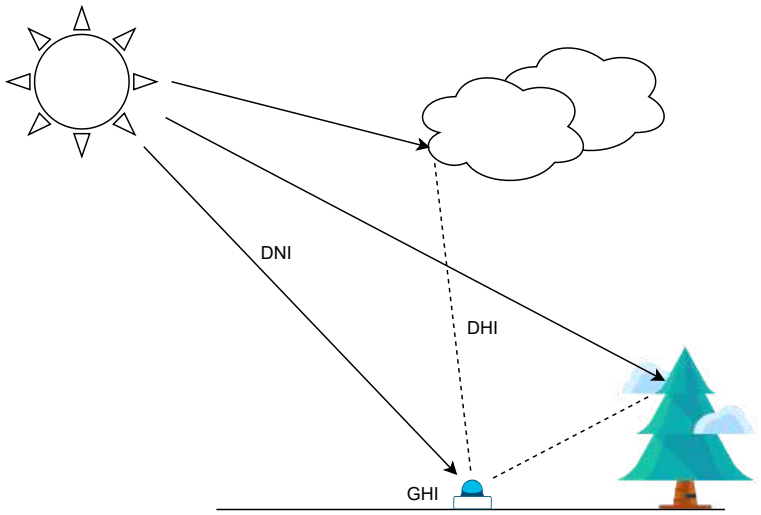


Figure 2.4: Components of the irradiance that reaches the Earth's surface. Adapted from [14].

The combination of two components, DNI and DHI, contributes to Global Horizontal Irradiance (GHI). In this way, GHI can be defined as the total solar energy received on the Earth's surface [8]. Fig. 2.4 provides a visual representation of the key components of solar radiation. The unit of measurement of GHI, DNI, and DHI is typically watts per square meter (W/m^2). This unit measures the power of solar radiation received per unit area. The relationship between these components can be expressed mathematically as:

$$GHI = DHI + DNI \cos(\Theta_z) \tag{2.6}$$

2.3 Physical models

In this section, the formulation of the physical models considered in this work is outlined. These models are used as benchmarks, providing a basis for the assessment of the performance and capabilities of ML-based synthetic data generation methods.

2.3.1 Collares-Pereira

The hourly solar radiation is determined based on the daily irradiance (H_T), the hour angle (ω), and the sunset hour angle (ω_s). In this model, the coefficients a and b are functions of ω_s and are included to account for the incidence angle effect [9]:

$$I = \frac{\pi H_T}{24}\left(\frac{\cos(\omega)-\cos(\omega_s)}{\omega_s\cdot\cos(\omega_s)-\sin(\omega_s)}\right)\cdot(a+b\cdot\cos(\omega)) \tag{2.7}$$

where the coefficients a and b are defined as follows:

$$a = 0.4090 + 0.5016\ \sin(\omega_s - 60^\circ), \tag{2.8a}$$

$$b = 0.6609 - 0.4767\ \sin(\omega_s - 60^\circ) \tag{2.8b}$$

2.3.2 Garg

To improve the accuracy of hourly solar radiation estimates during any season, an additional term has been introduced into the equation. This additional term helps improve the suitability of the model for estimating global radiation across different seasons [16]. The complete equation is given by

$$I = H_T\left(\frac{\pi}{24}\left(\frac{\cos(\omega)-\cos(\omega_s)}{\omega_s\cdot\cos(\omega_s)-\sin(\omega_s)}\right)-0.008\sin\left(3\left(\omega-0.65\right)\right)\right) \tag{2.9}$$

2.3.3 Yao

Yao proposed a model based on the consideration of solar geometric data to take into account different climatic conditions [47]. The equation is given by

$$I = \frac{\pi H_T}{24}\left(0.4762+0.6347\cos\omega\right)\frac{\left(\frac{24}{\pi}\sin\frac{\pi}{24}\cdot\cos\omega-\cos\omega_s\right)}{\sin\omega_s-\omega_s\cos\omega_s} \tag{2.10}$$

2.4 ML models

In this section, a brief overview of the models considered in this study, i.e., MLP, LSTM, GRU, and CNN, is provided. The main aspects and formulations for each model are detailed below.

2.4.1 Multi-layer perceptron

The well-known MLP is an artificial neural network capable of handling intricate patterns and solving challenging problems such as classification and regression. Usually referred to as universal approximators, MLPs can estimate any continuous function given enough hidden neurons and an appropriate training process [2]. Unlike a single perceptron, which has limitations in solving complex problems, it consists of multiple interconnected layers of neurons [15]. The output of a hidden neuron is determined by a linear combination of weights and inputs, followed by an activation function. Also, the input could be the output from the previous layer. Mathematically, the output of a neuron within a certain layer can be expressed as

$$y_i^l = \varphi^l \left(\sum_{j=1}^{n_{l-1}} W_{i,j}^l y_j^{l-1} + b_i^l \right), \tag{2.11}$$

where $l \in \{1, \dots, L\}$ corresponds to the hidden layer, $i \in \{1, \dots, n_l\}$ is the neuron, $W_{i,j}^l$ represents the weights, b_i^l is the bias term, and $\varphi^l(\cdot)$ is the non-linear activation function.

The non-linearity of activation functions allows neural networks to learn complex patterns and relationships in data that linear models cannot capture. By introducing non-linearity, these functions enable neural networks to approximate a wide range of functions, making them capable of solving more sophisticated problems and modeling real-world phenomena with greater accuracy. Common types of activation functions are sigmoid, hyperbolic tangent, relu, and softmax.

Backpropagation is a fundamental algorithm used to train feed-forward neural networks. The process begins with forward propagation, where input data passes through the network and produces predictions. The difference between the predicted output and the actual target is measured using a loss function. Backpropagation then applies the chain rule of calculus to compute the gradient of this loss with respect to each weight, layer by layer, starting from the output and moving backward. These gradients are used to update the weights in the direction that reduces the loss, typically through an optimization algorithm like gradient descent. By iteratively repeating this process over multiple training examples, the model refines its parameters, improving its ability to generalize to new data. This iterative process is called training, and the objective is to minimize the loss function.

2.4.2 Long short term memory

The LSTM is a type of RNN that can hold information for an extended time period. Fig. 2.5 illustrates the memory cell, which is the core component of an LSTM network, highlighting its capability to regulate information flow through various gates. To create a multi-layer LSTM network, several memory cells are stacked vertically. The sequence data is processed by each LSTM layer, and its output is passed to the next LSTM layer at each timestep. Finally, the output from the last LSTM layer is fed into a fully connected (Dense) layer to produce the final result. The equations that describe how an LSTM cell processes input data, updates its internal state, and produces an output at each time step are as follows [22]:

$$i_t = \varphi(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2.12a}$$

$$f_t = \varphi(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{2.12b}$$

$$o_t = \varphi(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{2.12c}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh(w_C \cdot [h_{t-1}, x_t] + b_C), \tag{2.12d}$$

$$h_t = o_t \cdot \tanh(C_t), \tag{2.12e}$$

where t represents the current time step; x_t is the input; $\varphi(\cdot)$ is the non-linear activation function; W_λ and b_λ, $\lambda \in \{i, f, o\}$ are the weights and biases of the input, forget, and output gates, respectively. Also, the hidden state is given by h_j, whereas the cell state C_j represents the memory of the LSTM network.

Recurrent networks, including LSTMs and GRUs, are trained using an algorithm called Backpropagation Through Time (BPTT) [44]. This algorithm is an extension of traditional backpropagation, specifically adapted to handle the sequential nature of RNNs. BPTT enables the network to learn from temporal dependencies in the data by "unfolding" the recurrent network across time steps. This unfolding process conceptually transforms the RNN into a deep feedforward network, where each layer represents a time step in the sequence. By doing so, BPTT can calculate how each parameter affects the output over the entire sequence. However, this approach introduces challenges, such as increased memory requirements for processing long sequences and the risk of vanishing or exploding gradients as time steps increase.

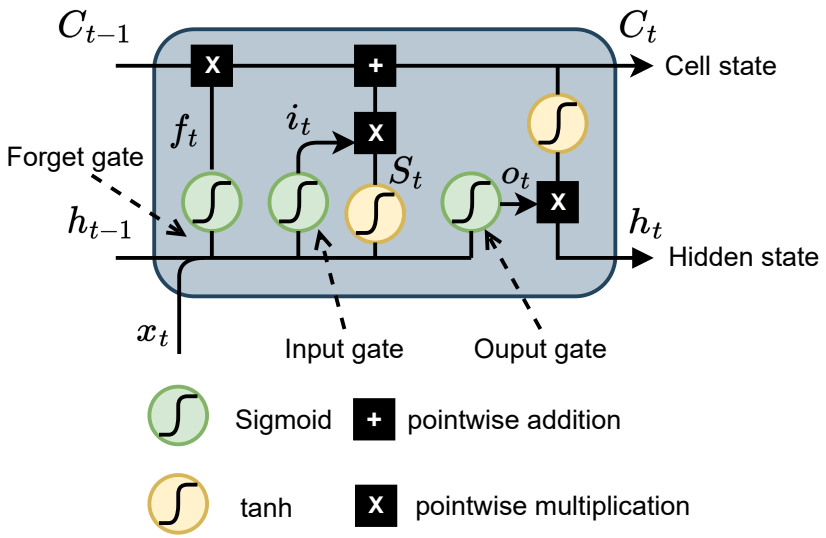


Figure 2.5: Structure of an LSTM cell. Recovery from [11].

2.4.3 Gate recurrent unit

The architecture of the GRU shares similarities with the LSTM model. Both networks employ gating mechanisms to dynamically adjust how much past information should be retained and how much new information should be incorporated at each time step [7]. Unlike LSTM, which typically has three gates, GRU uses only two gates and a different mechanism to update the hidden state. The behaviour of the GRU network is given by the following equations [6]

$$z_t = \varphi\left(W_z \cdot [h_{t-1}, x_t]\right), \tag{2.13a}$$

$$r_t = \varphi\left(W_r \cdot [h_{t-1}, x_t]\right), \tag{2.13b}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, \tag{2.13c}$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]), \tag{2.13d}$$

where W_λ, $\lambda \in \{z, r, h\}$, represents the weights of the update gate, reset gate, and hidden state, respectively. Also, h_t represents the current hidden state, $\tilde{h}_t$ symbolizes the current memory content weighted by the update gate, and $\odot$ denotes element-wise multiplication.

2.4.4 Convolutional neural network

This network automatically learns hierarchical features from raw input, eliminating the need for manual feature engineering [29]. Since we aim to model sequential data, in this work we focus on 1D CNNs. In a 1D CNN, a 1D kernel slides over the sequence capturing local patterns.

A kernel is a small vector of weights with shape $\mathbf{w} \in \mathbb{R}^L$, where $L \in \{3, 5, 7, ...\}$, and corresponds to the length of the kernel. The kernel computes the dot products with local regions of the input, and the result of this operation is a feature map. Multiple kernels are applied to capture various features from the input. The output of the first convolutional layer can be described by

$$y_i = \varphi\left(\sum_{j=0}^{L-1} \mathbf{w}_j \cdot \mathbf{x}_{i+j} + b\right), \tag{2.14}$$

where, y_i is the output at position i, L is the length of the kernel, $\mathbf{x}$ is the input vector, $\mathbf{w}$ is the kernel (weight) vector, b is the bias term, and $\varphi(\cdot)$ is the non-linear activation function. The most commonly used activation function in CNNs is ReLU (Rectified Linear Unit).

The size of the output feature map is influenced by the stride and padding used in the convolution layer. The stride refers to the steps size by which the kernel is shifted along the input data, with larger strides resulting in smaller output feature maps. For example, a stride of 1 means the kernel moves one step at a time, while a stride of 2 moves two steps at a time. Padding is a technique to add extra values (typically zeros) around the border of the input data. It helps control the size of the output feature map and preserve information at the edges of the input.

The pooling layer is a key component typically inserted after the convolutional layer. Its primary functions are to reduce the size of the sequence and decrease the computational load on the network. Pooling achieves this by summarizing information from local regions into single values. The two most common types of pooling are max pooling and average pooling.

In a deep CNN, convolutional layers are stacked sequentially, forming a hierarchical feature extractor. The initial layers typically detect low-level features such as edges and textures, while deeper layers progressively capture more complex, high-level features like shapes and object parts. Interspersed between these convolutional layers are pooling layers, most commonly max pooling, which serve to downsample the feature maps.

Chapter 3

Methodology

This chapter introduces the methodology adopted to evaluate the ML models. The steps outlined include a summary of the overall methodology, followed by detailed descriptions of the dataset, pre-processing techniques, and hyperparameters used in the evaluation process.

3.1 Overview

To improve clarity, the methodology steps for evaluating and optimizing the ML models are graphically represented in Fig. 3.1. A detailed explanation of each step is presented in the upcoming subsections.

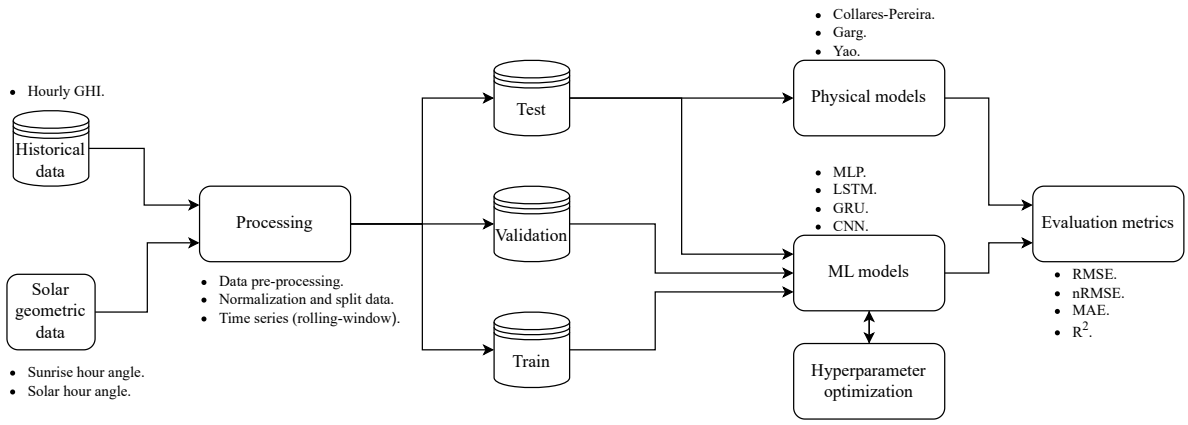


Figure 3.1: Overview of the methodology for evaluating ML models.

As illustrated in the flowchart in Fig.3.1, the process begins with a dataset containing real measurements of hourly GHI, as detailed in Section 3.2. Following this, the necessary pre-processing steps to prepare the data for analysis are described in Section 3.3. During this

stage, a daily dataset is derived from the available hourly data, and solar geometric data are incorporated into this dataset. By the end of this stage, a comprehensive dataset with daily irradiance (H_T), sunrise hour angle (ω_s), and hour angle (ω) variables representing the input features is obtained, along with the hourly GHI measurements that represent the output of the models. Next, the dataset is normalized and restructured using the rolling-window approach, as explained in Section 3.4. This is followed by splitting the dataset into three subsets: training, validation, and testing, as outlined in Section 3.5. In this section, the architecture of the models are also defined and the hyperparameter optimization is expounded in Section 3.6. Finally, the input features are fed into the ML models to downscale the data back to hourly. The output are then compared with the actual hourly GHI to evaluate the models' performance, using error metrics such as RMSE, nRMSE, MAE, and R^2.

3.2 Dataset

In this study, we make use of the data provided by INMET, available in [25]. It comprises hourly measurements of various meteorological parameters, including: global radiation, total hourly rainfall, atmospheric pressure at station level, air temperature, dew point temperature, humidity levels, and wind characteristics. For the purpose of our analysis, we have centered on the Brasilia meteorological station, located at Latitude: $15.78^{\circ}S$ and Longitude: $47.92^{\circ}O$, with an elevation of 1161 meters. Additionally, we have selectively utilized data spanning from January 1, 2017, to December 31, 2023, covering a 7-year period. This selection results in a total of 61344 data points. Table 3.1 summarize this information.

Table 3.1: Detailed information about the dataset.

Station	Latitude (°S)	Longitude (°O)	Elevation (m)	Time period	Resolution	Data points
Brasilia	15.78	-47.92	1161	Jan, 2017 - Dec, 2023	1 h	61344

3.3 Data processing

The data pre-processing stage is crucial in many fields, including ML [46], since it can significantly improve the performance and efficiency of models. In this work, it involves the application of various techniques and methodologies to ensure the accuracy and reliability of

solar radiation data[36, 21]. The procedures applied to GHI data include visual inspection, handling missing values, implementing quality constraints, and removing outliers. As a first step, we visualize the data to identify any significant issues. Typically, this involves plotting the measurements over time and generating histograms. This preliminary qualitative analysis aims to detect major problems within the dataset. In the Fig. 3.2, the left plot illustrate the raw dataset, allowing us to observe trends and anomalies over time. The right plot is a histogram of the data, which helps us understand the distribution of values and identify any skewness.

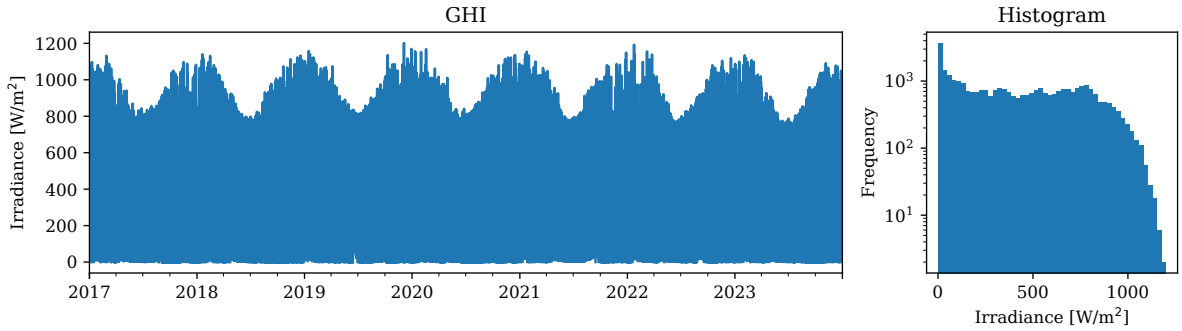


Figure 3.2: Raw GHI data available from Jan/2017 to Dec/2023 and histogram.

Fig. 3.3 shows a short period of GHI measurements in a 2-Dimensional plane where each pixel represents samples from the dataset with the intensity color indicating the measurement value. Here, the yellow dashed line indicates sunrise and sunset in the local area which was calculated using the solar position algorithm introduced by [37] and available in [24]. The figure depicts a GHI data gap on January 3, 2018, as well as low values of GHI before the sunrise and after the sunset. To avoid these minor variations, GHI values before the sunrise and after the sunset were set to zero. The figure also shows that measurements neither commence nor conclude in alignment with the sunrise and sunset, respectively. Consequently, the dataset exhibits a positive offset of one hour.

Now, to identify outliers and possible erroneous measurements in our dataset, we apply the test recommended by [31]. It consists of comparing the measurements with two bounds that can be reached by the correct global irradiance values. The bounds for GHI are given by

$$\beta < GHI < \alpha\, H_0\, Cos^{1.2}(\Theta_z) + \Delta, \tag{3.1}$$

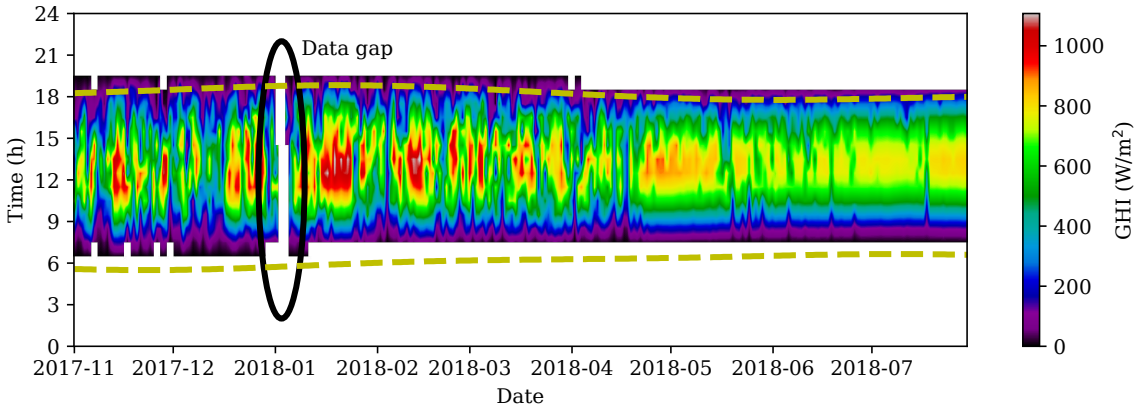


Figure 3.3: GHI data with each data point visible for identifying gaps, time shifts, and missing data.

where $\beta \in \{-2, -4\}$, $\alpha \in \{1.5, 1.2\}$, $\Delta \in \{100, 50\}$, H_0 is the extraterrestrial irradiance calculated using the Spencer method presented in [40], and Θ_z is the solar zenith angle. When $\beta = -4$, $\alpha = 1.5$ and $\Delta = 100$ it is referred to as physically possible limit; whereas for $\beta = -2$, $\alpha = 1.2$ and $\Delta = 50$ as a extremely rare limit.

After applying the test [31] over our dataset, measurements that are outside the physically possible and extremely rare limits were found for $75^\circ < \Theta_z < 90^\circ$ and correspond to samples taken during sunrise and sunset. Specifically, the three measurements that surpassed the physical possible limit were replaced by null values whereas the 482 measurements that exceeded the extremely rare limit were replaced by the upper bound in (3.1). Fig. 3.4 shows the filtered data.

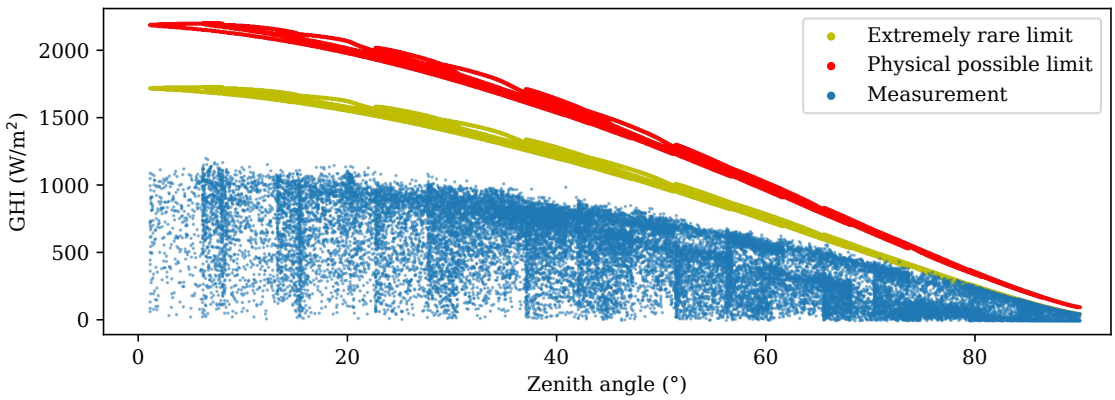


Figure 3.4: Quality control bounds for hourly GHI measurements.

The last step in the dataset cleaning process involves linear interpolation to fill missing data points (up to 3 consecutive points), and the removal of entire days if any data gap exists. Fig. 3.5 illustrates the impact of quality control procedures on the dataset, demonstrating

consistent data, corrected time shifts, and data filtered between the sunrise and sunset time period. The resulting dataset, after all the data pre-processing tasks, consists of 61272 data points with hourly resolution.

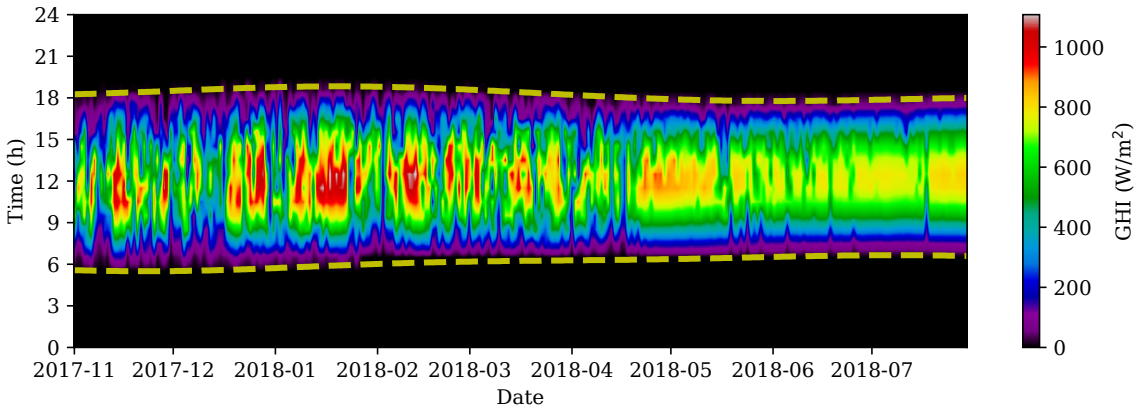


Figure 3.5: Graphical representation of the resulting dataset after the data pre-processing stage.

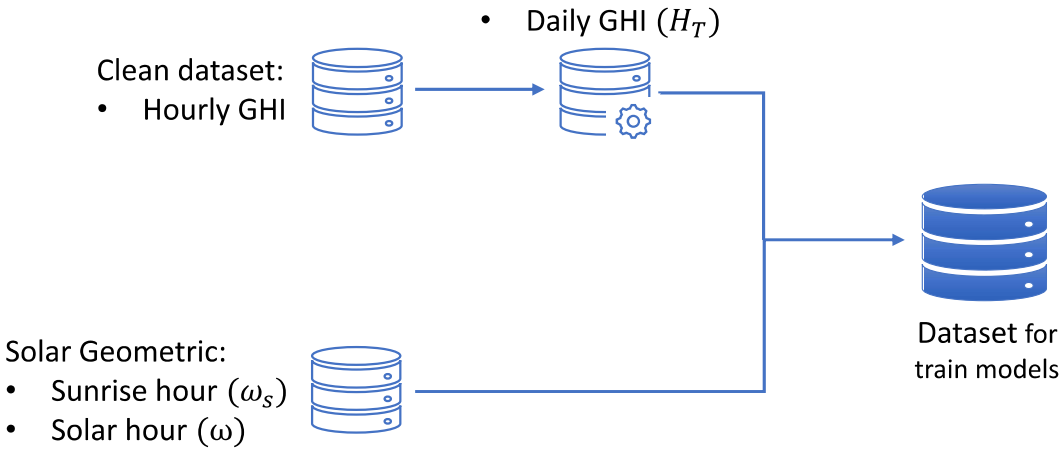


Figure 3.6: Process flow for creating the training dataset.

The focus of this work is on generating synthetic hourly data based on daily GHI measurements. To achieve this, Fig 3.6 illustrates the entire process of creating the dataset used to train ML models. It begins with the hourly GHI data, which is processed and cleaned. Next, the daily iiradiance (H_T) was calculated by integrating the hourly GHI measurements throughout the day using the trapezoidal rule. Additionally, the sunrise hour angle (ω_s) and solar hour angle (ω), both in radians, were included in the dataset. Here ω_s was calculated using the solar position algorithm presented in [37] while ω is given by

$$\omega = (T - 12)\,\frac{\pi}{12}, \tag{3.2}$$

where T is the local solar time (in hours). It is important to mention that the solar hour angle, ω, is not added directly to the dataset because it varies throughout the day. Fig. 3.7 depicts an example, showing the dimension inconsistency between the variables. In Fig. 3.7a, H_T and ω_s have a single value, whereas ω has 24 distinct values. To address this issue, we used the resample and forward-fill methods available in [42]. With this method we guarantee that the H_T and ω_s dimension match with ω as shown in Fig. 3.7b. This adjustment addresses this dimensional mismatch by aligning all variables.

$$\begin{bmatrix} H_T^0 & \omega_S^0 \end{bmatrix} \begin{bmatrix} \omega^0 \\ \omega^1 \\ \omega^2 \\ \vdots \\ \omega^{23} \end{bmatrix} \qquad \begin{bmatrix} H_T^0 & \omega_S^0 & \omega^0 \\ H_T^0 & \omega_S^0 & \omega^1 \\ H_T^0 & \omega_S^0 & \omega^2 \\ \vdots & \vdots & \vdots \\ H_T^0 & \omega_S^0 & \omega^{23} \end{bmatrix}$$

a) b)

Figure 3.7: Data matrix for a single day. a) Data vectors with different sizes. b) Equal-sized data.

3.4 Rolling Window Time Series Structure

The time series using a rolling-window approach allows capturing temporal dependencies and patterns, making it valuable for time series forecasting. In this work, we explore this approach specifically to generate hourly synthetic GHI data. The procedure involves dividing the time series data into smaller segments or windows, where each window contains a fixed number of data points. By feeding these overlapping windows into a model, new data points can be generated based on past observations.

For illustrative purposes, Fig. 3.8 depicts the process of creating one-sample of a time series. More specifically, given a data set $\boldsymbol{D} \in \mathbb{R}^{m \times k}$ with m observations and k features, we define a window size (n) and horizon (h). The former represents a sequence of consecutive lag observations as input for the model, while the latter is the forecast period into the future for which predictions are made. Note that the window size can be variable to control the quantity of past information that is fed in the model. After defining the values of n and h, the shape of the input data is given by $x \in \mathbb{R}^{r \times n \times k-1}$, and the output data has the shape $y \in \mathbb{R}^{r \times h}$, where r is the number of training samples or batch size.

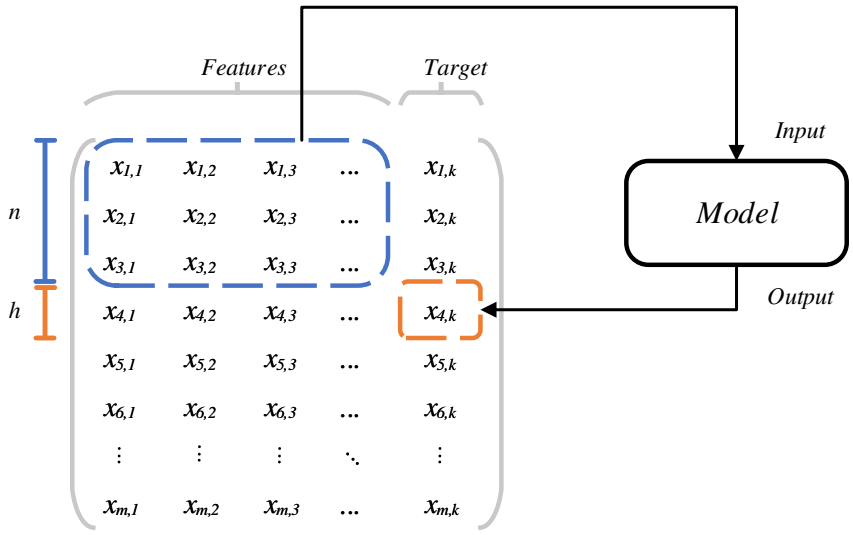


Figure 3.8: Rolling-window approach for $n = 3$ and $h = 1$.

3.5 Evaluation Methodology

At this stage, the data have been pre-processed, and a dataset with H_T, ω_s, and ω feature inputs was created, as explained in Section 3.3. This dataset has an hourly resolution, meaning there are 24 intervals per day. The data were normalized using the Min-Max normalization technique to scale the values within the range of 0 to 1. The equation is given by

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.3}$$

where x is the original value, x_{norm} is the normalized value, $\min(x)$ and $\max(x)$ represent the minimum and maximum values of the feature, respectively.

The normalized data was then chronologically split into three subsets: training, validation, and testing. With 7 years of available data, the first 5 years were used for training, the subsequent year for validation, and the final year for testing. This split corresponds to approximately 72% for training, 14% for validation, and 14% for testing. These subsets were restructured using past samples to estimate the next hourly GHI value (horizon = 1). The past samples include daily irradiance (H_T), sunrise (ω_s), and hour angle (ω), but do not incorporate past hourly GHI values, as explained in Section 3.4, This approach focuses on estimation based on related features rather than direct forecasting from previous GHI values. Hence, the shape of the input matrix are defined as $r \times n \times 3$ and the output as $r \times 1$, where r is the batch size, n is the windows size, and 3 corresponds to the three features (H_T, ω_s, and ω).

In this work, the values of r and n are defined as hyperparameters, with their optimal values being determined through hyperparameter tuning.

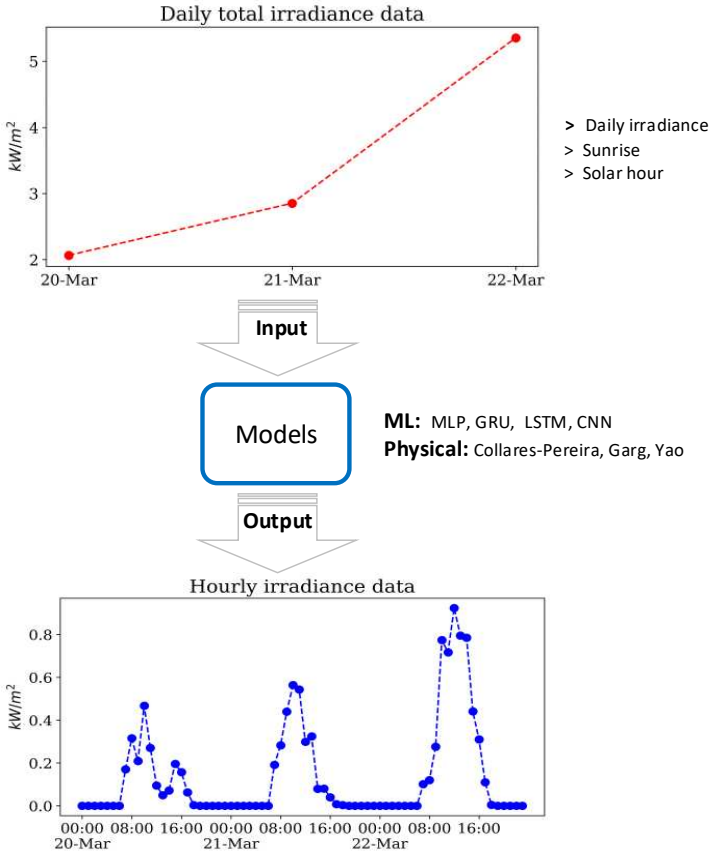


Figure 3.9: Evaluation methodology

Fig. 3.9 illustrated the process to estimate hourly GHI values using various ML approaches such as MLP, LSTM, GRU, and CNN. For LSTM and GRU networks, the input shape remains $r \times n \times 3$. This shape is ideal for these recurrent networks, which process data step by step along the sequence dimension (windows size). For CNN, the input shape is also maintained as $r \times n \times 3$, treating the sequence dimension as a 1D spatial dimension with 3 channels. However, for the MLP network, the input data is flattened, resulting in a 2D shape of $r \times (n * 3)$, where each sample in the batch is represented as a single vector of length $(n * 3)$, combining all features across the entire sequence.

For comparison, also are included physical models such as Collares-Pereira, Garg, and Yao. Before beginning the training process, the architecture of each model is defined, as show Fig. 3.10. Here, the MLP, GRU, and LSTM models each contain two hidden layers with dropout regularization, followed by a fully connected layer. The CNN model employs a six-layer configuration, with three 1D convolutional layers, each followed by a max-pooling

layer. The output from the final layer is flattened into a 1D vector, to which dropout is then applied. Regarding the physical models, the expressions outlined in Section 2 were used for evaluation. These models rely on established mathematical formulations and do not require parameter tuning or training. Consequently, physical models can generate hourly data directly from the test set without the need for additional preprocessing, such as normalization or rolling windows.

The outputs of the models are the estimated GHI values for each hour, which are then compared to the actual GHI values to assess the performance and accuracy of each model. The metrics used for this evaluation include RMSE, nRMSE, MAE, and the R^2 metric. RMSE provides a measure of the average magnitude of the errors, nRMSE normalizes this error to make it comparable across different scales, MAE gives the average absolute difference between the estimated and actual values, and the R2 metric indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. These metrics can be calculated as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(O_i - P_i)^2}{n}}, \tag{3.4a}$$

$$\text{nRMSE} = \frac{\text{RMSE}}{O_{max}}, \tag{3.4b}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|O_i - P_i|}{n}, \tag{3.4c}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1} n(O_i - P_i)^2}, \tag{3.4d}$$

where O_i represents the i-th true measurement, P_i represents the i-th output value of the models n is the total number of data points, and O_{max} is the maximum value present in true measurement.

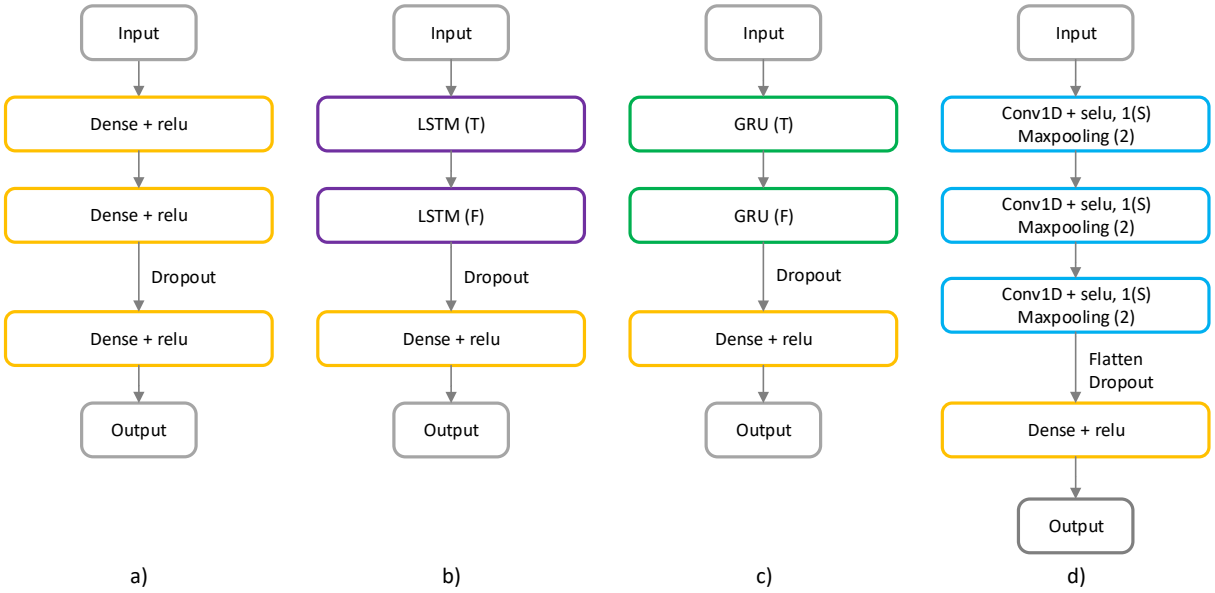


Figure 3.10: Architecture of ML models a) MLP b) LSTM c) GRU d) CNN

3.6 Hyperparameters optimization

To obtain the best accuracy from the ML models, an optimization of its hyperparameters was conducted. Initially, we defined a search space containing all possible values for the hyperparameters. Subsequently, we employed random search to randomly sample a fixed number of configurations from the search space to train the models [3]. The number of iterations was set to 30, meaning that 30 different configurations of the model were randomly generated and evaluated. Each iteration represents a unique set of hyperparameters, creating distinct initializations of the model. This approach ensures a diverse exploration of the hyperparameter space, allowing us to assess a wide range of model configurations. The optimal model is then selected based on the lowest RMSE value achieved on the validation set. The optimization task was executed with *weights and biases* python library [4] and the hyperparameter's search space is detailed in Table 3.2. For all models, common hyperparameters such as window size, batch size, and learning rate were optimized. Also, for MLP, GRU, and LSTM the size of the two hidden layers, dropout rate, and size of the fully connected layer were refined. Finally, for the CNN, the filter size, kernel size, dropout rate, and size of the fully connected layer were fine-tuned.

The models were trained using the Adam optimizer, and the mean square error was used as a loss function. The training process was set to run for 20 epochs. To ensure optimal model performance, convergence was monitored by tracking the training and validation losses. A model was considered converged when these indicators showed consistent stability

Table 3.2: Configuration of search space

Model	Hyperparameter	Range
All	windows size	[1, 3, 6, 12, 18, 24, 48, 72]
	batch size	[32, 64, 256, 512, 1024]
	learning rate	[5, 3, 1, 0.5, 0.3]*
	last layer	[32, 128, 32**]
MLP,	dropout	[0.1, 0.3]
LSTM,	layer 1	[32, 256, 32**]
GRU.	layer 2	[32, 256, 32**]
CNN	filters 1	[16, 64, 16**]
	filters 2	[16, 128, 16**]
	filters 3	[16, 128, 16**]
	kernels 1	[3, 5, 9, 13]
	kernels 2	[3, 5, 7]
	kernels 3	[3, 5, 7]

* Expressed in scientific notation: $\times 10^{-3}$

** Quantization step size.

over multiple epochs. The learning curves illustrating how the loss function evolves across epochs for both the training and validation datasets are provided in Appendix C. Additionally, early stopping was implemented to prevent overfitting and reduce unnecessary computations with a patience parameter set to 10. This means that training would be stopped early if the validation loss did not improve for 10 consecutive epochs, preserving the best model encountered during the training process.

Chapter 4

Results and discussion

This chapter presents the main results obtained from Chapter 3 jointly with a detailed analysis of the impact of the input features is presented.

4.1 Impact of window size

This analysis focuses on the impact of window size on the model's performance among optimized hyperparameters. Our findings indicate that the optimal window size does not necessarily correspond to the largest number of past samples. In fact, a continuous increase in window size eventually leads to a decrease in performance. Conversely, overly small window sizes do not provide enough context for models to perform adequately. Therefore, the ideal window size is best determined through a hyperparameter optimization process. Table 4.1 presents the performance metrics for various window sizes with the optimal one, for each model, highlighted in bold. The optimal window size for the MLP, LSTM, and GRU models was 12, while the best window size for the CNN model was 72. The table also indicates that the LSTM architecture yields the best overall performance with this optimal window size. Interestingly, the MLP model achieves the second-best performance in terms of RMSE, outperforming the GRU model despite the latter's architectural similarity to the LSTM. Given its simpler design, the MLP model can be a compelling alternative to more complex architectures, offering a balance between accuracy and model simplicity.

In addition to window size, other hyperparameters were fine-tuned using the random search technique. Table 4.2 summarizes the optimal set of hyperparameters for each model evaluated in this study. The optimal configurations varied among the different models,

Table 4.1: Error metrics in test set for different window sizes and for each ML model. The bold number shows the best metric values.

Model	Window size	RMSE (W/m^2)	nRMSE (%)	MAE (W/m^2)	R^2
MLP	1	80,201	7,070	39,933	0,933
	6	80,637	7,108	39,778	0,932
	12	**80,047**	**7,056**	**40,677**	**0,933**
	24	80,562	7,102	39,763	0,932
	48	80,561	7,102	41,497	0,932
	72	81,994	7,228	39,843	0,930
LSTM	1	81,036	7,143	40,682	0,931
	6	80,125	7,063	41,357	0,933
	12	**79,967**	**7,049**	**38,486**	**0,933**
	24	80,150	7,065	39,443	0,933
	48	80,929	7,134	40,562	0,931
GRU	1	80,882	7,130	39,233	0,931
	6	80,294	7,078	39,287	0,932
	12	**80,210**	**7,071**	**39,257**	**0,933**
	24	80,466	7,093	39,996	0,933
	48	80,391	7,086	42,045	0,931
CNN	6	80,504	7,096	39,077	0,932
	12	80,773	7,120	40,620	0,932
	24	80,309	7,079	40,034	0,932
	48	80,772	7,120	39,375	0,932
	72	**80,107**	**7,061**	**39,166**	**0,933**
	96	80,986	7,139	40,598	0,931

Table 4.2: Optimal hyperparameter values for each ML model

Hyperparameter	MLP	LSTM	GRU	CNN
windows size	12	12	12	72
batch size	32	64	64	512
learning rate	0.001	0.0003	0.0003	0.0005
layer last	128	128	96	224
dropout	0.1	0.3	0.1	-
layer 1	32	192	128	-
layer 2	128	64	32	-
filters 1	-	-	-	32
filters 2	-	-	-	64
filters 3	-	-	-	80
kernels 1	-	-	-	3
kernels 2	-	-	-	7
kernels 3	-	-	-	5

reflecting the unique equations that drive each network's internal dynamics. These findings underscore the critical role that hyperparameter tuning plays in the performance of ML models.

4.2 Influence of input features

In this subsection, we examine the impact of each input feature on the estimation of hourly GHI using the MLP model with its optimal hyperparameters, as defined in the previous subsection. The MLP strikes an optimal balance between accuracy and computational speed, making it ideal for a comprehensive investigation of feature importance. Its efficiency allows for faster analysis of three scenarios without significant delay. The three scenarios consist of the following input features: (i) H_T, ω_s, and ω, representing the scenario evaluated thus far; (ii) H_T and ω; and finally (iii) H_T and ω_s. The performance of each scenario, based on the given metrics, is presented in Table 4.3. Additionally, Fig. 4.1 depicts the estimated hourly GHI values for each of them. It's noteworthy that the worst performance occurs for scenario (ii), which uses H_T and ω_s as inputs. This poor outcome is likely due to these features remaining constant throughout the day, causing the model to struggle with capturing the daily solar radiation profile. In scenario (iii), which uses H_T and ω as input features, every performance metric shows significant improvement. This enhancement is likely due to the

Table 4.3: Error metrics using MLP network with n=12 and varying the input features.

Input features	RMSE (W/m^2)	nRMSE (%)	MAE (W/m^2)	R^2 -
H_T, ω, ω_s	80,047	7,056	40,677	0,933
H_T, ω	82,823	7,301	42,798	0,928
H_T, ω_s	236,105	20,813	173,589	0,416

H_T is the total daily irradiance, ω is the solar hour angle and ω_s is the sunrise hour angle

model's ability to associate the solar hour profile with the corresponding solar radiation trends. Additionally, this scenario suggests that the amplitude of the GHI curve at solar noon is primarily influenced by H_T. Finally, when the inputs are H_T, ω_s, and ω, (corresponding to scenario (i)), the MLP model achieves its highest performance in estimating hourly GHI values. This configuration allows the model to capture a broader range of information, leading to more accurate predictions.

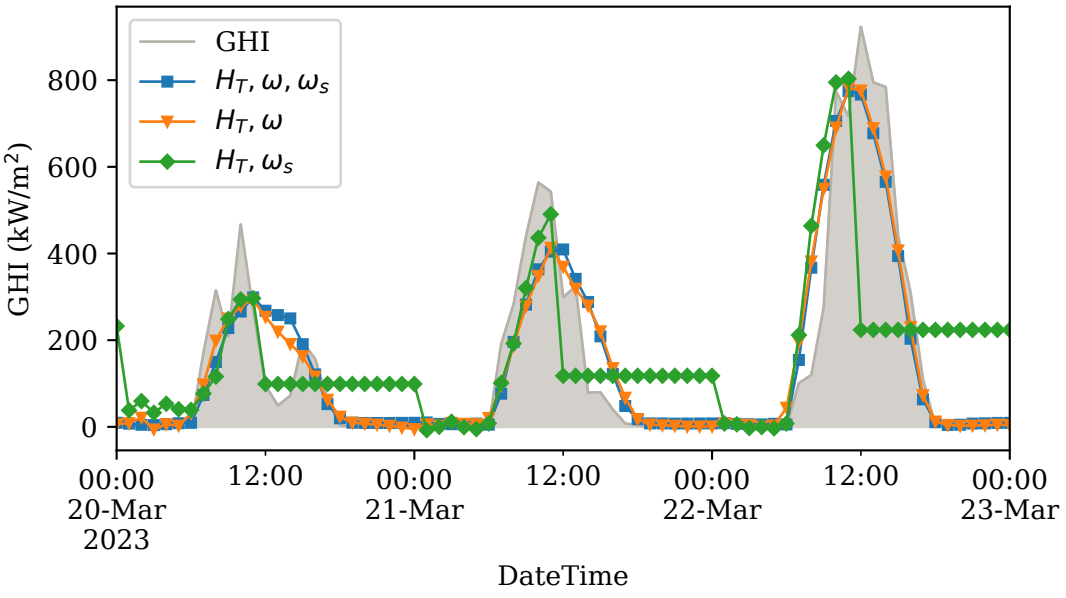


Figure 4.1: Estimated hourly GHI values using MLP network with n=12 and varying the input features.

4.3 Performance of the optimized models

To assess the accuracy and benefits of our data-driven approach, we compared optimized ML models with the three physical models described in Section 2.3. We used the same test set to

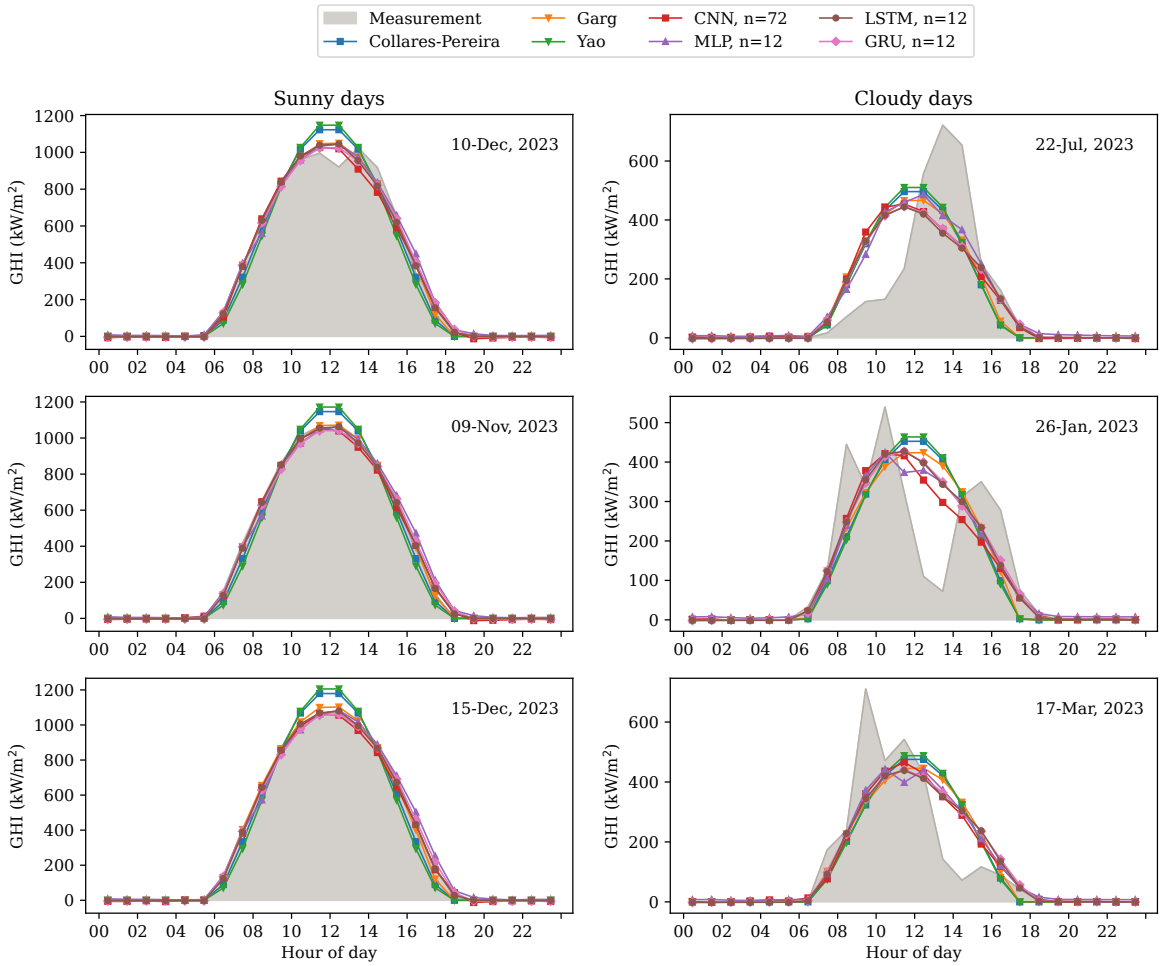


Figure 4.2: Estimated hourly GHI values generated by all evaluated models and true measured under two extreme scenarios: sunny days and cloudy days.

evaluate both ML and physical models. Table 4.4 presents the performance of each model, sorted from the lowest to the highest error rate. The results indicate that all ML models outperformed the physical models, with the LSTM model achieving the best overall performance. On average, the LSTM model improves the performance of the least effective model, referred to as Yao, by 15.12% in terms of RMSE. When compared to the Collares-Pereira and Garg physical models, LSTM outperforms them by 12.05% and 7.68%, respectively. It is worth mentioning that the table displays performance metrics for ML models with optimized window sizes. However, even when considering non-optimal window sizes, the ML models consistently outperformed the physical models. These findings underscore the robustness and adaptability of ML approaches compared to traditional physical modeling techniques.

To further illustrate the effectiveness of the previously discussed models, Fig. 4.2 presents the estimated hourly GHI values from each model, compared against actual GHI measurements

Table 4.4: Error metrics for ML models and physical models. Only models with the optimal window size are considered.

Model	RMSE (W/m^2)	nRMSE (%)	MAE (W/m^2)	R^2 -
LSTM, n=12	79,967	7,049	38,486	0,933
MLP, n=12	80,047	7,056	40,677	0,933
CNN, n=72	80,107	7,061	39,166	0,933
GRU, n=12	80,210	7,071	39,257	0,933
Garg	86,625	7,636	42,828	0,921
Collares-Pereira	90,924	8,015	46,991	0,913
Yao	94,220	8,306	49,542	0,907

for two different weather conditions: sunny and cloudy days. The figure demonstrates that all models successfully capture the general GHI profile in both conditions. Notably, it also highlights the ability of the ML models to adapt to seasonal variations in time. For example, every ML model accurately tracked the GHI profile on November 9, when low GHI values were recorded at 6:00, and on March 17, when no GHI values were observed at 6:00.

When examining the sunny day samples—the optimal scenario for estimating hourly GHI values—we find that the ML models consistently track the true measurements throughout the day. In contrast, physical models tend to overestimate GHI, especially during the solar noon period. This overestimation is most pronounced in the Collares-Pereira and Yao models, which show significant deviations from the actual GHI profile at solar noon. Among the physical models, Garg's approach stands out as the top performer, exhibiting the least overestimation and generally providing more accurate estimates throughout the day compared to the other physical models.

Fig. 4.3 shows the estimated GHI obtained by all evaluated models. The estimates from the ML models align more closely with the true measurement profile than those from the physical models. Consequently, ML models handle uncertainties, primarily caused by cloud cover and varying climatic conditions throughout the day, more effectively than physical models.

Regarding performance under cloudy conditions, the physical models generate a smoother curve, as expected, since their equations lack explicit variables for modeling adverse atmospheric conditions like cloud cover. In this scenario, the ML models demonstrate superior flexibility in adjusting their predictions, allowing them to cope with

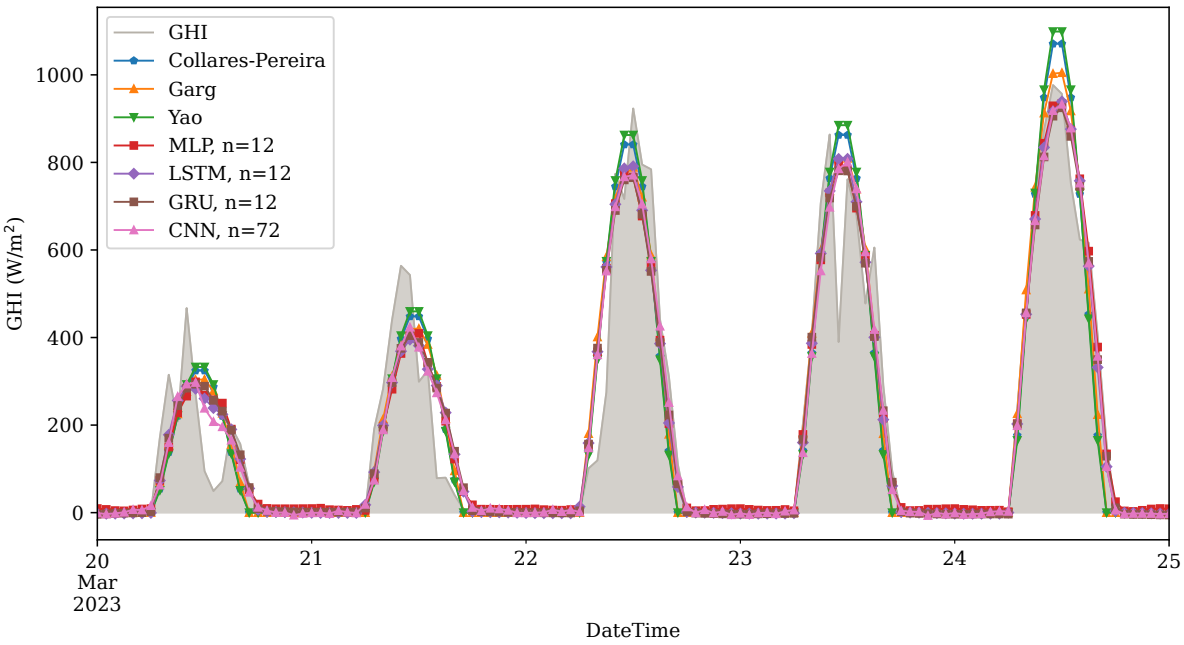


Figure 4.3: Estimated hourly GHI values generated by all evaluated models for 5 days.

dynamic weather conditions. This adaptability results in the LSTM network outperforming all other models under both sunny and cloudy conditions.

Chapter 5

Conclusions and Future Works

5.1 Conclusion

This work introduces an novel approach to GHI data downscaling using memory-based ML models, including LSTM, GRU, and a 1D CNN. The approach requires only daily GHI measurements, the sunrise hour angle, and the solar hour angle as input parameters, without reliance on detailed meteorological data. The results indicate that these ML models outperform the well-known physical models of Collares-Pereira, Garg, and Yao. According to the comprehensive performance evaluation conducted in this study, the LSTM model achieved the lowest error metrics among all evaluated models. Additionally, this work underscores the advantages of applying hyperparameter optimization to improve the performance of ML models. Among the physical models considered, the Garg model demonstrated the best results, outperforming the Collares-Pereira and Yao models. This study not only establishes the superiority of memory-based ML models over traditional physical models but also emphasizes the value of hyperparameter tuning in enhancing model accuracy.

5.2 Future works

Propose a machine learning model for synthesizing a one-minute global irradiance time series based on hourly averaged data as input. By training on historical data, the model learns to downscale from hourly averages into high-resolution minute-by-minute values, providing a detailed and precise representation of solar irradiance fluctuations. Additionally,

the model can been extended to predict other relevant solar irradiance components, including direct normal irradiance and diffuse horizontal irradiance,which are essential for comprehensive solar energy analysis.

Explore generative networks and attention mechanisms to generate high-resolution solar data. These approaches can be particularly effective in creating realistic minute-level solar irradiance data from hourly averages. Generative adversarial networks consist of a generator and a discriminator that work together to produce high-quality synthetic data. Attention mechanisms, on the other hand, enhance the model's ability to capture temporal dependencies and patterns in the data. By focusing on relevant parts of the input sequence, attention mechanisms enable the model to better understand the relationships between different time steps.

To ensure that the synthetic solar irradiance data accurately replicates the statistical properties of real-world measurements, distribution metrics can be utilized to quantify the similarities or differences between the ground truth data and the generated data. Metrics such as the Kullback-Leibler divergence, Jensen-Shannon divergence, and Kolmogorov-Smirnov integration provide a comprehensive assessment of how well the distributions of the synthetic data align with those of the actual measurements. By comparing the probability distributions of key variables like global irradiance, direct normal irradiance, and diffuse horizontal irradiance, the fidelity of the synthetic data will be evaluated.

Bibliography

[1] R. Aguiar, M. Collares-Pereira, and J. Conde. "Simple procedure for generating sequences of daily radiation values using a library of Markov transition matrices". In: *Solar Energy* 40.3 (1988), pp. 269–279. ISSN: 0038-092X. DOI: https://doi.org/10.1016/0038-092X(88)90049-7. URL: https://www.sciencedirect.com/science/article/pii/0038092X88900497.

[2] J.-G. Attali and G. Pagès. "Approximations of Functions by a Multilayer Perceptron: a New Approach". In: *Neural Networks* 10.6 (1997), pp. 1069–1081. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(97)00010-5. URL: https://www.sciencedirect.com/science/article/pii/S0893608097000105.

[3] J. Bergstra and Y. Bengio. "Random search for hyper-parameter optimization." In: *Journal of machine learning research* 13.2 (2012).

[4] L. Biewald. *Experiment Tracking with Weights and Biases*. 2020. URL: https://www.wandb.com/.

[5] J. Bright, C. Smith, P. Taylor, and R. Crook. "Stochastic generation of synthetic minutely irradiance time series derived from mean hourly weather observation data". In: *Solar Energy* 115 (2015), pp. 229–242. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2015.02.032. URL: https://www.sciencedirect.com/science/article/pii/S0038092X15001024.

[6] K. Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[8] L. D. N. Coelho. "Modelos de estimativa das componentes de radiação solar a partir de dados meteorológicos". Português. In: (Mar. 2017). Accepted: 2017-03-27T17:04:13Z. DOI: 10.26512/2016.11.D.23091. URL: http://repositorio2.unb.br/jspui/handle/10482/23091 (visited on 06/30/2024).

[9] M. Collares-Pereira and A. Rabl. "The average distribution of solar radiation-correlations between diffuse and hemispherical and between daily and hourly insolation values". In: *Solar Energy* 22.2 (Jan. 1979), pp. 155–164. ISSN: 0038-092X. DOI: 10.1016/0038-092X(79)90100-2. URL: https://www.sciencedirect.com/science/article/pii/0038092X79901002 (visited on 04/03/2024).

[10] J. C. Cortez et al. “Fuzzy Ensemble Algorithm for Day-ahead Photovoltaic Power Forecasting.” In: *2024 International Conference on Smart Energy Systems and Technologies (SEST)*. [Accepted for publication].

[11] J. C. Cortez Aucapiña. “Day-ahead photovoltaic power forecasting based on a hybrid deep learning methodology [recurso eletrônico] Previsão de geração fotovoltaica para o dia seguinte baseado em uma metodologia hibrida de aprendizado profundo Juan Carlos Cortez Aucapiña”. eng. In: (Jan. 2023). Publisher: [s.n.] URL: https://research.ebsco.com/linkprocessor/plink?id=35334cd2-f15f-3d71-b2ca-fcc7b42ec33d.

[12] J. A. Cumbicos, L. P. Jimenéz Jimenéz, G. Fraidenraich, and T. Barros. “Hourly GHI Data Estimation from Daily Measurements Using Machine Learning Techniques.” In: *IEEE Open Access Journal of Power and Energy* (2024). [Under review].

[13] *Extraterrestrial Radiation | Solar Radiation*. URL: https://www.greenrhinoenergy.com/solar/radiation/extraterrestrial.php (visited on 06/04/2024).

[14] *Forecasting Solar Radiation using DataRobot to Optimize Power Generation | DataRobot AI Cloud*. en-US. URL: https://www.datarobot.com/blog/forecasting-solar-radiation-using-datarobot-to-optimize-power-generation/ (visited on 07/01/2024).

[15] M. W. Gardner and S. R. Dorling. “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences”. In: *Atmospheric Environment* 32.14 (1998), pp. 2627–2636. ISSN: 1352-2310. DOI: https://doi.org/10.1016/S1352-2310(97)00447-0. URL: https://www.sciencedirect.com/science/article/pii/S1352231097004470.

[16] H. P. Garg and S. N. Garg. “Improved correlation of daily and hourly diffuse radiation with global radiation for Indian stations”. In: *Solar & Wind Technology* 4.2 (Jan. 1987), pp. 113–126. ISSN: 0741-983X. DOI: 10.1016/0741-983X(87)90037-3. URL: https://www.sciencedirect.com/science/article/pii/0741983X87900373 (visited on 04/03/2024).

[17] M. M. Gilbert. *Renewable and efficient electric power systems*. John Wiley & Sons, 2004.

[18] V. Graham and K. Hollands. “A method to generate synthetic hourly solar radiation globally”. In: *Solar Energy* 44.6 (1990), pp. 333–341. ISSN: 0038-092X. DOI: https://doi.org/10.1016/0038-092X(90)90137-2. URL: https://www.sciencedirect.com/science/article/pii/0038092X90901372.

[19] C. A. Gueymard. “Revised composite extraterrestrial spectrum based on recent solar irradiance observations”. In: *Solar Energy* 169 (2018). Publisher: Elsevier, pp. 434–440.

[20] M. A. Hassan, M. Abubakr, and A. Khalil. “A profile-free non-parametric approach towards generation of synthetic hourly global solar irradiation data from daily totals”. In: *Renewable Energy* 167 (2021), pp. 613–628. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2020.11.125. URL: https://www.sciencedirect.com/science/article/pii/S0960148120318772.

[21] M. A. Hassan, A. Khalil, and M. Abubakr. "Selection methodology of representative meteorological days for assessment of renewable energy systems". In: *Renewable Energy* 177 (2021), pp. 34–51. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2021.05.124. URL: https://www.sciencedirect.com/science/article/pii/S0960148121008077.

[22] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". English. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[23] M. Hofmann et al. "Improved Synthesis of Global Irradiance with One-Minute Resolution for PV System Simulations". In: *International Journal of Photoenergy* 2014 (Nov. 2014). Ed. by P. H. Borse. Publisher: Hindawi Publishing Corporation, p. 808509. ISSN: 1110-662X. DOI: 10.1155/2014/808509. URL: https://doi.org/10.1155/2014/808509.

[24] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski. "pvlib python: a python package for modeling solar energy systems". In: *Journal of Open Source Software* 3.29 (2018). Publisher: The Open Journal, p. 884. DOI: 10.21105/joss.00884. URL: https://doi.org/10.21105/joss.00884.

[25] *Instituto Nacional de Meteorologia - INMET*. URL: https://portal.inmet.gov.br/dadoshistoricos (visited on 02/19/2024).

[26] M. Iqbal. *An Introduction To Solar Radiation*. Elsevier Science, 2012. ISBN: 978-0-323-15181-8. URL: https://books.google.com.br/books?id=3_qWce_mbPsC.

[27] H. D. Kambezidis. "The Solar Resource". In: *Comprehensive Renewable Energy (Second Edition)*. Ed. by T. M. Letcher. Oxford: Elsevier, Jan. 2022, pp. 26–117. ISBN: 978-0-12-819734-9. DOI: 10.1016/B978-0-12-819727-1.00002-9. URL: https://www.sciencedirect.com/science/article/pii/B9780128197271000029 (visited on 06/03/2024).

[28] T. Khatib and W. Elmenreich. "A model for hourly solar radiation data generation from daily solar radiation data using a generalized regression artificial neural network". In: *International Journal of Photoenergy* 2015 (2015), pp. 1–13. DOI: 10.1155/2015/968024.

[29] C. .-. J. Kuo. "Understanding convolutional neural networks with a mathematical model". In: *Journal of Visual Communication and Image Representation* 41 (Nov. 2016), pp. 406–413. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2016.11.003. URL: https://www.sciencedirect.com/science/article/pii/S1047320316302267 (visited on 03/30/2024).

[30] T. Letcher. "Comprehensive renewable energy". In: (2022).

[31] C. N. Long and E. G. Dutton. *BSRN Global Network recommended QC tests, V2. x*. Publisher: PANGAEA. 2010.

[32] A. Luque and S. Hegedus. *Handbook of photovoltaic science and engineering*. John Wiley & Sons, 2011.

[33] L. Mora-López and M. Sidrach-de-Cardona. "Multiplicative ARMA models to generate hourly series of global irradiation". In: *Solar Energy* 63.5 (1998), pp. 283–291. ISSN: 0038-092X. DOI: https://doi.org/10.1016/S0038-092X(98)00078-4. URL: https://www.sciencedirect.com/science/article/pii/S0038092X98000784.

[34] B. Ngoko, H. Sugihara, and T. Funaki. "Synthetic generation of high temporal resolution solar radiation data using Markov models". In: *Solar Energy* 103 (2014), pp. 160–170. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2014.02.026. URL: https://www.sciencedirect.com/science/article/pii/S0038092X14001042.

[35] A. G. Olabi. "RENEWABLE ENERGY-VOLUME 1: SOLAR, WIND, AND HYDROPOWER DEFINITIONS, DEVELOPMENTS, APPLICATIONS, CASE STUDIES, AND MODELLING AND SIMULATION". In: (2023). Publisher: Elsevier.

[36] D. Perez-Astudillo, D. Bachour, and L. Martin-Pomares. "Improved quality control protocols on solar radiation measurements". In: *Solar Energy* 169 (2018), pp. 425–433. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2018.05.028. URL: https://www.sciencedirect.com/science/article/pii/S0038092X18304614.

[37] I. Reda and A. Andreas. "Solar position algorithm for solar radiation applications". In: *Solar Energy* 76.5 (Jan. 2004), pp. 577–589. ISSN: 0038-092X. DOI: 10.1016/j.solener.2003.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0038092X0300450X (visited on 02/26/2024).

[38] M. K. d. Silva. "Estudo de modelos matemáticos para análise da radiação solar e desenvolvimento de ferramenta para modelagem e simulação de sistemas fotovoltaicos [recurso eletrônico] Michelle Kitayama da Silva". por. In: (Jan. 2019). Publisher: [s.n.] URL: https://research.ebsco.com/linkprocessor/plink?id=412fbf9a-b526-392a-8449-490448627e2e.

[39] *SONDA - Sistema Nacional de Organização de Dados Ambientais*. URL: https://sonda.ccst.inpe.br/ (visited on 06/27/2024).

[40] J. Spencer. "Fourier series representation of the position of the sun." In: *Search* 2.5 (1971), p. 172.

[41] J. Stein, C. Hansen, and M. Reno. *Global horizontal irradiance clear sky models : implementation and analysis*. en. Tech. rep. SAND2012-2389, 1039404. Mar. 2012, SAND2012–2389, 1039404. DOI: 10.2172/1039404. URL: https://www.osti.gov/servlets/purl/1039404/ (visited on 05/23/2024).

[42] T. p. d. team. *pandas-dev/pandas: Pandas*. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[43] K. Vidyanandan. "An Overview of Factors Affecting the Performance of Solar PV Systems". In: *Energy Scan (A house journal of Corporate Planning, NTPC Ltd.)* 27 (Feb. 2017), pp. 2–8.

[44] P. J. Werbos. “Backpropagation through time: what it does and how to do it”. In: *Proceedings of the IEEE* 78.10 (1990). Publisher: IEEE, pp. 1550–1560.

[45] A. Whillier. “The determination of hourly values of total solar radiation from daily summations”. In: *Archiv für Meteorologie, Geophysik und Bioklimatologie Serie B* 7.2 (Mar. 1956), pp. 197–204. DOI: 10.1007/bf02243322.

[46] D. Yang, G. M. Yagli, and H. Quan. “Quality Control for Solar Irradiance Data”. In: *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*. 2018, pp. 208–213. DOI: 10.1109/ISGT-Asia.2018.8467892.

[47] W. Yao et al. “New decomposition models to estimate hourly global solar radiation from the daily value”. In: *Solar Energy* 120 (Oct. 2015), pp. 87–99. ISSN: 0038-092X. DOI: 10.1016/j.solener.2015.05.038. URL: https://www.sciencedirect.com/science/article/pii/S0038092X15002984 (visited on 04/03/2024).

Appendix A

Data exploration

This appendix presents an overview of the data exploration process. Fig. A.1 depicts a heat map spanning from 2017 to 2023, revealing temporal patterns across a 24-hour daily cycle. The visualization highlights consistent daily cycles and annual trends, with peak irradiance typically occurring around midday. These patterns, along with the pre-processing steps taken to address any data irregularities, ensure the dataset is thoroughly prepared for model input.

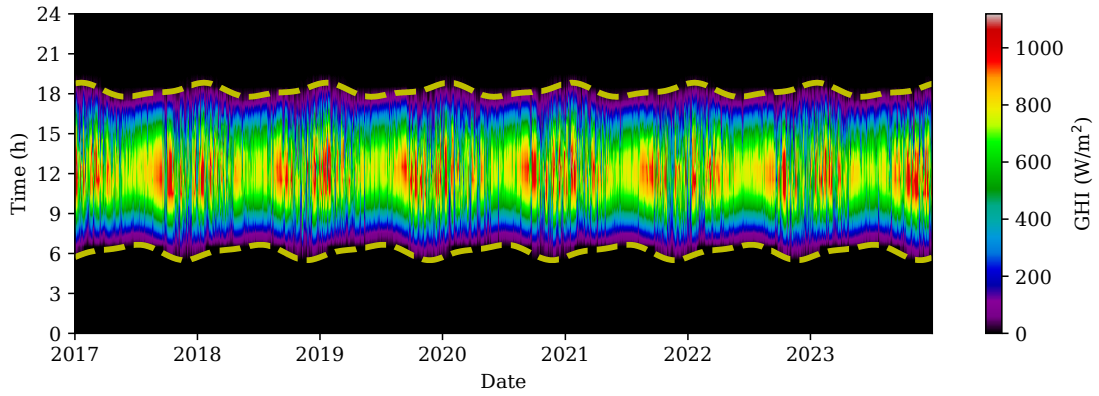


Figure A.1: Resulting dataset after applying data pre-processing.

Fig. A.2 presents a series of plots depicting the daily GHI profiles for one representative day each month in 2023. Each subplot illustrates the variation in GHI throughout a single day, highlighting the seasonal changes in solar radiation. The figure shows longer and more intense irradiance periods during the summer months (e.g., July and August) compared to the shorter, less intense periods in the winter months (e.g., January and December). This visualization effectively represents the daily solar cycle and its seasonal fluctuations, providing crucial context for model testing.

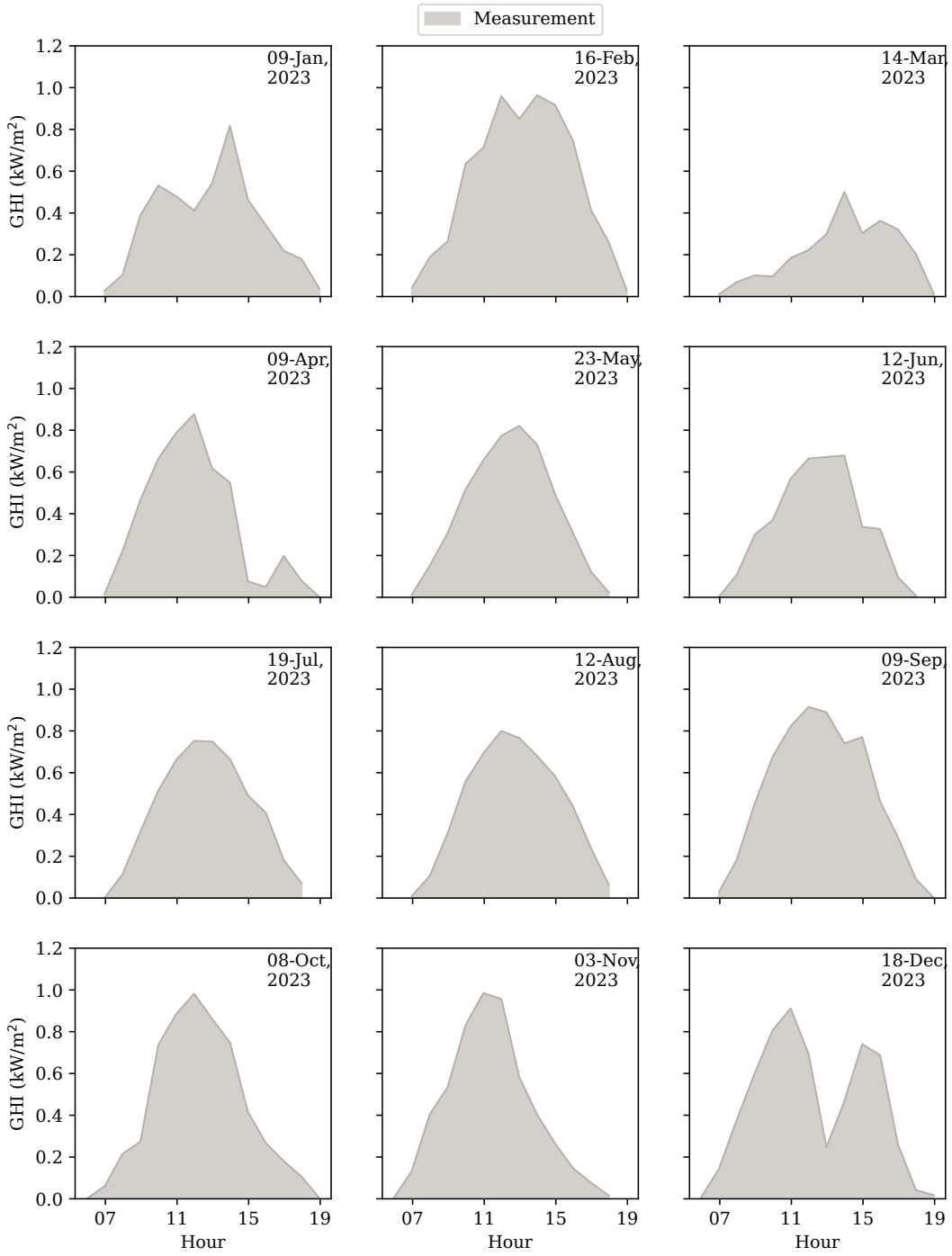


Figure A.2: Sample day for each month of the test set.

Appendix B

Hyperparameter Optimization

For each ML model and window size, we conducted a hyperparameter optimization process based on the search space defined in Table 3.2. This appendix provides a summary of the optimization results for the LSTM model. Given the multiple model runs, we utilized a figure to compare the experiments and visualize their accuracy with different hyperparameters. Fig. B.1 illustrates the random combinations explored to achieve the best performance.

Tables B.1-B.5 show the top five configurations ranked according to their RMSE values. The errors shown in these tables were obtained from the validation set. Only the model with the minimum error was selected for application to the test set.

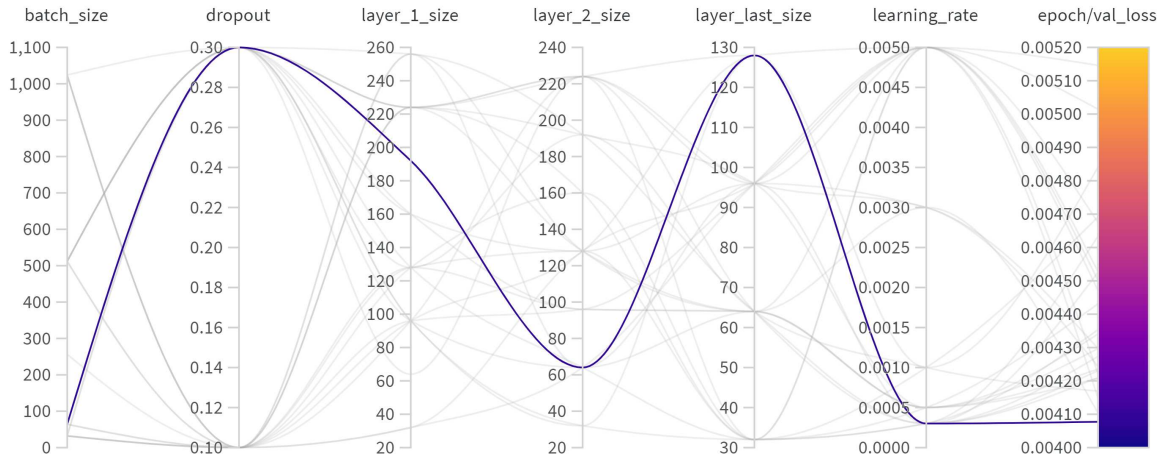


Figure B.1: All hyperparameter configurations tested for LSTM with window size = 12. The blue line indicates the best configuration.

Table B.1: Top 5 hyperparameters configuration for LSTM and windows size = 1.

ID	RMSE	MAE	batch_size	dropout	layer_1	layer_2	layer_last	learning_rate	# exp
1	0,06455	0,03500	512	0,1	96	96	128	0,003	27
2	0,06474	0,03359	64	0,1	64	128	32	0,003	7
3	0,06476	0,03449	256	0,3	224	256	96	0,005	17
4	0,06489	0,03666	1024	0,3	128	192	96	0,005	25
5	0,06508	0,03643	1024	0,1	224	160	96	0,003	4

Table B.2: Top 5 hyperparameters configuration for LSTM and windows size = 6.

ID	RMSE	MAE	batch_size	dropout	layer_1	layer_2	layer_last	learning_rate	# exp
1	0,06400	0,03277	256	0,1	224	96	96	0,0005	16
2	0,06429	0,03257	1024	0,1	64	128	32	0,005	8
3	0,06430	0,03276	1024	0,3	64	224	64	0,005	19
4	0,06464	0,03248	32	0,1	192	128	128	0,001	21
5	0,06471	0,03123	64	0,3	160	128	64	0,0003	1

Table B.3: Top 5 hyperparameters configuration for LSTM and windows size = 12.

ID	RMSE	MAE	batch_size	dropout	epochs	layer_1	layer_2	layer_last	learning_rate	# exp
1	0,06385	0,03048	64	0,3	20	192	64	128	0,0003	11
2	0,06406	0,03111	32	0,1	20	224	128	64	0,003	25
3	0,06441	0,03127	32	0,1	20	32	64	96	0,005	2
4	0,06457	0,03114	512	0,3	20	128	192	64	0,001	9
5	0,06487	0,03133	32	0,3	20	160	96	64	0,0005	23

Table B.4: Top 5 hyperparameters configuration for LSTM and windows size = 24.

ID	RMSE	MAE	batch_size	dropout	epochs	layer_1	layer_2	layer_last	learning_rate	# exp
1	0,06409	0,03105	1024	0,1	20	224	160	64	0,003	11
2	0,06425	0,03035	64	0,3	20	64	128	32	0,0003	5
3	0,06427	0,03185	64	0,1	20	224	192	64	0,0003	14
4	0,06428	0,03105	32	0,3	20	256	160	64	0,003	2
5	0,06446	0,03106	32	0,3	20	160	64	96	0,001	14

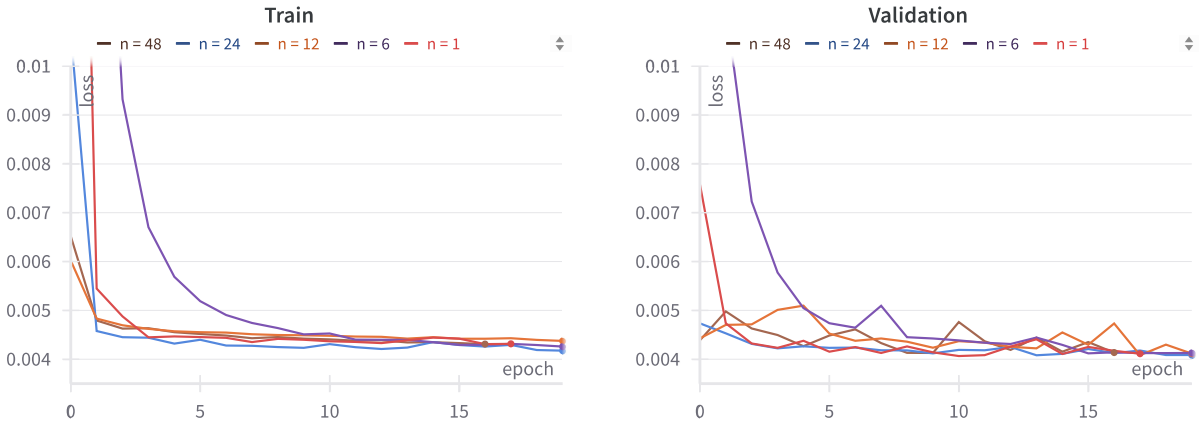
Table B.5: Top 5 hyperparameters configuration for LSTM and windows size = 48.

ID	RMSE	MAE	batch_size	dropout	epochs	layer_1	layer_2	layer_last	learning_rate	# exp
1	0,06420	0,03087	512	0,1	20	96	160	96	0,005	4
2	0,06448	0,03124	32	0,1	20	96	160	96	0,0003	8
3	0,06467	0,03091	256	0,1	20	64	224	128	0,005	3
4	0,06498	0,03332	32	0,3	20	96	160	32	0,005	2
5	0,06531	0,03204	256	0,3	20	96	192	32	0,003	7

Appendix C

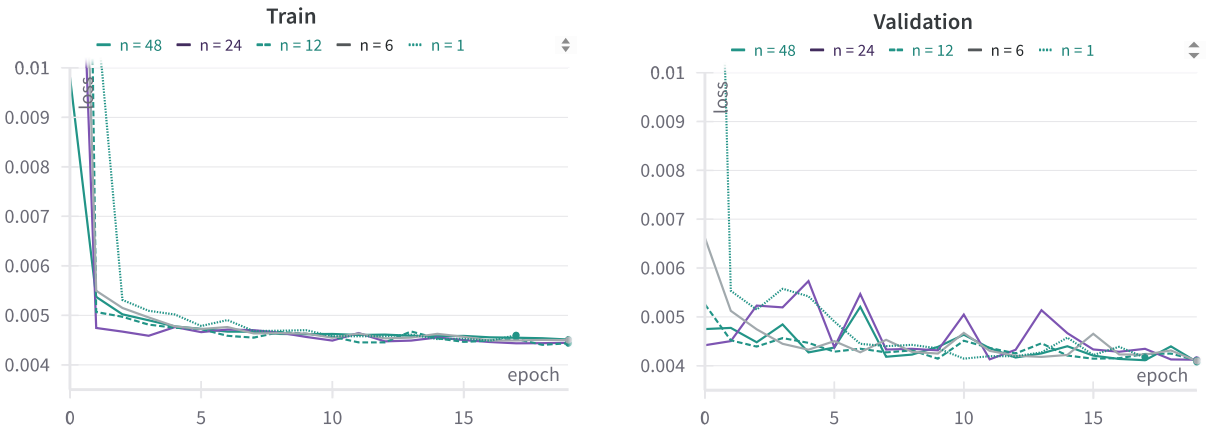
Learning curves

This appendix presents the learning curves for MLP, LSTM, GRU, and CNN models, illustrating the progression of the loss function during model training (see Figs C.1-C.4). The figures are organized by model type, with each graph displaying curves for different window sizes. These curves demonstrate how the MSE evolves across epochs for both the training and validation datasets. All models exhibit a general trend of decreasing loss over epochs, providing visual insight into their respective convergence processes. The validation curves, however, display more fluctuations than the training curves, particularly for models with smaller window sizes, suggesting increased sensitivity to specific data points in the validation set. After approximately 10 epochs, most models reach a relatively stable loss, with only minor variations thereafter.



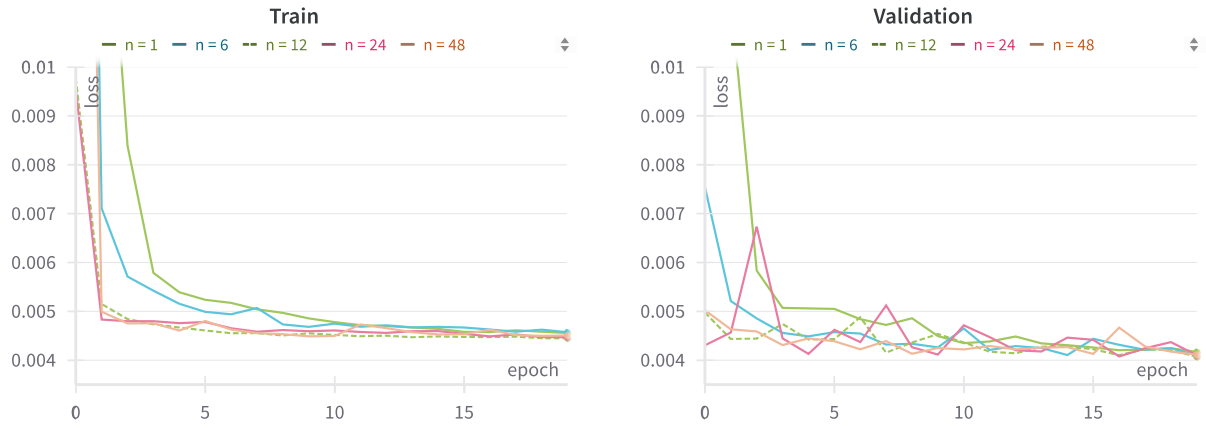
(a) (b)

Figure C.1: Learning curves for MLP and different window sizes.



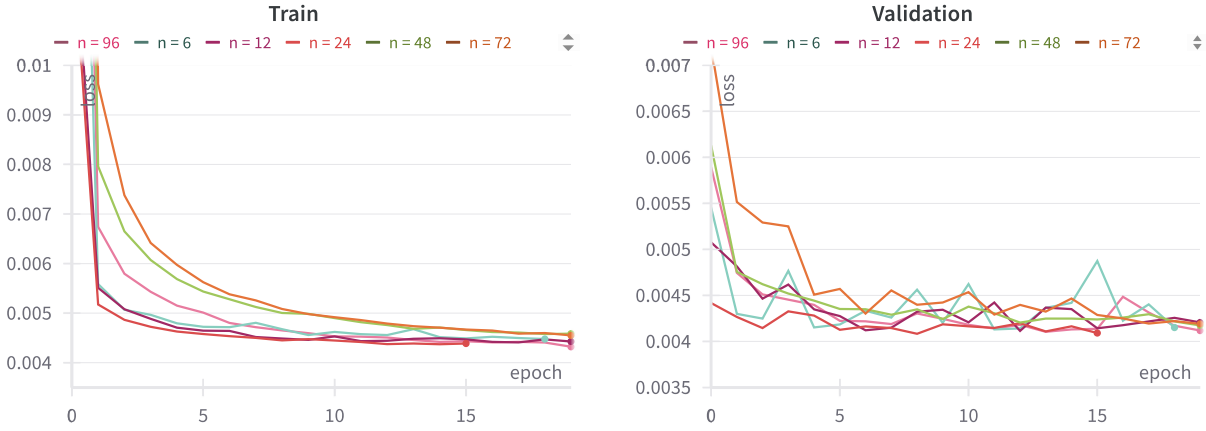
(a) (b)

Figure C.2: Learning curves for LSTM and different window sizes.



(a) (b)

Figure C.3: Learning curves for GRU and different window sizes.



(a) (b)

Figure C.4: Learning curves for CNN and different window sizes.