DESIGN OF MACHINE LEARNING MODELS FOR PREDICTING IRRADIANCE IN BIFACIAL SOLAR PANELS

JUAN MANUEL RODRIGUEZ SARMIENTO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2026

DESIGN OF MACHINE LEARNING MODELS FOR PREDICTING IRRADIANCE IN BIFACIAL SOLAR PANELS

JUAN MANUEL RODRIGUEZ SARMIENTO

Thesis submitted in partial fulfillment of the requirements for the degree in Electrical Engineering

Advisor:
César Antonio Duarte Gualdrón, PhD

Co-advisors:
Steven Hegedus, PhD
Luis Felipe Carreño Barrera, MSc candidate

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2026

Dedicated to my mother, my father, and my family.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my thesis advisors, César Duarte, Steven Hegedus, and Felipe Carreño, for their patience, passion, ambition, and dedication to this project. Each of them has achieved extraordinary things as an engineer, and I now deeply admire them. In addition, Fernando Bustos also helped me and was a source of inspiration and guidance for this project.

I sincerely thank the Industrial University of Santander (UIS) because it truly changed my life, made me a better person, gave me the confidence I needed from the beginning, and allowed me to achieve more dreams than I ever imagined. The sum of all my experiences at the university has made me the engineer I am today. I must confess that at first, I wasn't proud of myself, but after everything I've been through at the university, I couldn't be happier and prouder of the man I've become. Even if I had all the money in the world, I wouldn't change the opportunity to be part of UIS and live this wonderful experience that I will never forget.

I also express my sincere gratitude to all the professors and friends at the University, who always believed in me, even during difficult times. They supported me, taught me valuable lessons, and along this path, what I remember most are the hugs, the laughter, and the meaningful words I hold in my heart from my teachers and friends.

Finally, I want to thank God for guiding me on this path, giving me the wisdom and intelligence to make the best decisions at this stage of my life, and for showing me all His power, which I didn't deserve.

With appreciation, **Juan Manuel Rodriguez Sarmiento**

TABLE OF CONTENTS

LIST OF TABLES

Page.

LIST OF FIGURES

Page.

RESUMEN

TÍTULO: DISEÑO DE MODELOS DE APRENDIZAJE AUTOMÁTICO PARA PREDECIR LA IRRADIANCIA EN PANELES SOLARES BIFACIALES [*]

AUTOR: JUAN MANUEL RODRIGUEZ SARMIENTO [**]

PALABRAS CLAVE: Módulos Bifaciales, Predicción de irradiancia, Aprendizaje Profundo, Redes neuronales recurrentes, Redes neuronales convolucionales, Redes neuronales híbridas, Predicción de series temporales.

DESCRIPCIÓN: Uno de los principales desafíos que enfrenta la red eléctrica moderna es la integración de energías renovables caracterizadas por su variabilidad. Entre estas tecnologías, los módulos bifaciales han ganado relevancia debido a su capacidad de capturar irradiancia por ambas caras (frontal y trasera). Sin embargo, su integración efectiva requiere el desarrollo de modelos de predicción de irradiancia confiables.

En este contexto, el presente trabajo propone el desarrollo, evaluación y análisis de ablación de modelos de inteligencia artificial basados en aprendizaje profundo. Se analizan diferentes arquitecturas de redes neuronales, incluyendo redes neuronales recurrentes, convolucionales temporales y modelos híbridos, además del efecto de variables meteorológicas, métodos de escalado de datos y la integración de múltiples sensores en un mismo modelo.

Los resultados muestran que ciertas arquitecturas, como BiGRU y los modelos híbridos basados en CNN, presentan mayor estabilidad en la predicción de irradiancia. Asimismo, se identifican diferencias relevantes en el desempeño dependiendo del método de escalado y del tipo de modelo utilizado. Finalmente, se observa que en este proyecto R^2 resultó más útil que MAE para clasificar los días según su complejidad predictiva. Estos hallazgos contribuyen al desarrollo de metodologías robustas para la integración de sistemas fotovoltaicos bifaciales en redes eléctricas modernas.

[*] Trabajo de Grado

[**] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones. Director: Cesar Antonio Duarte Gualdron, PhD. Codirectores: Steven Hegedus, PhD; Luis Felipe Carreño Barrera, estudiante de maestría.

ABSTRACT

TITLE: DESIGN OF MACHINE LEARNING MODELS FOR PREDICTING IRRADIANCE IN BIFACIAL SOLAR PANELS [*]

AUTHOR: JUAN MANUEL RODRIGUEZ SARMIENTO [1]

KEYWORDS: Bifacial Modules, Irradiance Forecasting, Deep Learning, Recurrent Neural Networks, Convolutional Neural Networks, Hybrid Neural Networks, Time Series Prediction.

DESCRIPTION: Modern power systems face increasing challenges related to growing energy demand, grid stability, and the integration of renewable energy sources characterized by inherent variability. Among these technologies, bifacial photovoltaic modules have gained significant attention due to their ability to capture irradiance from both the front and rear surfaces, thereby increasing their energy generation potential. However, their effective integration requires the development of reliable irradiance forecasting models. In this context, this work proposes the development, evaluation, and ablation analysis of deep learning–based irradiance prediction models. Different neural network architectures are explored, including recurrent neural networks, temporal convolutional networks, and hybrid models. Additionally, the influence of meteorological variables, data scaling strategies, and the integration of multiple sensors within a single predictive model are analyzed.

The results show that certain architectures, such as BiGRU and CNN-based hybrid models, provide greater stability in irradiance prediction. Furthermore, relevant performance differences are observed depending on the scaling method and model structure. Finally, for this project, the R^2 metric is identified as more suitable than metrics such as MAE for identifying days with higher predictive complexity. These findings contribute to the development of more robust methodologies for integrating bifacial photovoltaic systems into Smart Grids environments.

[*] Degree Work

[1] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones. Advisor: Cesar Antonio Duarte Gualdron, PhD. Coadvisors: Steven Hegedus, PhD; Luis Felipe Carreño Barrera, MSc Candidate.

INTRODUCTION

The global energy system is undergoing a profound transformation driven by increasing electricity demand, the rapid expansion of digital technologies, and the urgent need to reduce greenhouse gas emissions. In this context, Smart grids have emerged as a key paradigm for modernizing electrical infrastructure, enabling the integration of renewable energy sources, improving system reliability, and facilitating more flexible and efficient energy management [5]. Within this transition, distributed renewable generation plays a central role, requiring not only new infrastructure but also advanced predictive tools capable of anticipating the behavior of energy resources under varying environmental conditions.

Among renewable technologies, photovoltaic solar energy has become one of the most prominent solutions due to its scalability and declining costs [1]. In recent years, bifacial photovoltaic modules have gained increasing attention because of their ability to capture solar radiation from both the front and rear surfaces, allowing them to harness direct, diffuse, and reflected irradiance [2]. This capability can significantly increase energy yield compared to traditional monofacial panels. However, the performance of bifacial systems strongly depends on several environmental and operational variables such as ground albedo, module spacing, tilt angle, temperature, humidity, and projected shadows [3], which introduce considerable variability into the energy production process.

Accurate forecasting of irradiance, therefore, becomes a critical requirement for the effective integration of bifacial photovoltaic systems into modern electrical networks. Traditional approaches based on physical or statistical models have been widely used for solar prediction [4], yet their performance often deteriorates when applied to complex or highly dynamic environments [1]. Furthermore, research on bifacial modules remains relatively limited compared to the extensive literature available for monofacial photovoltaic systems, particularly when real operational data are used.

Recent advances in machine learning offer promising alternatives for addressing these challenges. Neural network architectures designed for time-series analysis—such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), that belong to the category of recurrent neural networks (RNN), Bidirectional recurrent networks (BiLSTM and BiGRU), Convolutional Neural Networks (CNN), Temporal convolutional Networks (TCN), and hybrid

models that combine temporal and spatial feature extraction like CNN-LSTM or CNN-GRU—have demonstrated strong capabilities for capturing nonlinear relationships and temporal dependencies in complex datasets [1]. These characteristics make such models particularly suitable for forecasting solar irradiance, where environmental variability and sequential patterns play a significant role.

In order to rigorously assess the performance of the proposed forecasting models, it is necessary to define appropriate evaluation metrics that capture both error magnitude and predictive reliability. In this study, commonly used regression metrics are employed, including Mean Absolute Error (MAE), the coefficient of determination (R^2), and Mean Absolute Percentage Error (MAPE) [4]. These metrics provide complementary perspectives on model performance: MAE quantifies the average prediction error in absolute terms, R^2 evaluates the model's ability to explain the variance of the observed data, and MAPE offers a relative measure of error expressed as a percentage [4]. Establishing these metrics at this stage allows for a consistent and interpretable comparison of the different modeling approaches analyzed throughout this work.

To address the existing research gap, this work proposes the design and implementation of time-series-based machine learning models for irradiance forecasting in bifacial photovoltaic modules. Several neural network architectures are evaluated using real experimental data obtained from bifacial panel installations. The study also examines the influence of different meteorological variables, data scaling strategies, and the integration of multiple sensors within the predictive models.

The results demonstrate that advanced neural network architectures can significantly improve irradiance forecasting performance, particularly under complex environmental conditions. In addition, the study provides insights into model stability, feature contribution, and the suitability of different preprocessing strategies. These findings contribute to a better understanding of bifacial photovoltaic behavior and provide practical tools for their future integration into smart grid environments, supporting the transition toward more sustainable and reliable energy systems.

1. PROJECT OBJECTIVES

1.1 GENERAL OBJECTIVE

To design time-series-based machine learning models to analyze and predict rear irradiance in bifacial solar panels with a 24-hour forecast horizon to optimize energy production estimates and contribute to the stability and reliability of smart grids.

1.2 SPECIFIC OBJECTIVES

●Implement and train time-series forecasting models using architectures such as RNN, LSTM, CNN, and hybrid models to capture spatial and temporal patterns in irradiance.

●To evaluate the contribution of input variables to forecasting models through ablation analysis, quantifying the effect of removing individual variables or groups of variables on predictive performance to support model interpretability and feature selection.

●Document the study's conclusions, highlighting its contribution to the understanding of bifacial panels and their future integration into smart grids.

2. IRRADIANCE PREDICTION IN BIFACIAL MODULES

This chapter introduces the problem of irradiance prediction in bifacial photovoltaic modules and the main variables involved in the capture of both front and rear irradiance. This chapter also highlights the growing importance of this topic and the need to establish a stronger scientific foundation for bifacial technology beyond the simple goal of increasing energy production. Although research specifically focused on rear irradiance prediction remains limited, numerous studies have addressed front-side irradiance forecasting and the simulation of bifacial energy production. These contributions not only show that neural networks have been an effective solution to address the irradiance forecasting problem but also establish a baseline for the present project. All of this, together with prior research on the operation of bifacial panels, is reviewed to provide the necessary context for understanding the phenomenon and the motivation behind the proposed solution.

2.1 PROBLEM STATEMENT AND RELEVANCE

Bifacial modules have become a growing trend in the solar energy industry due to their potential to increase energy yield compared to conventional monofacial systems [6]. Unlike monofacial modules, which capture irradiance only from the front surface, bifacial modules can receive radiation from both the front and rear sides, converting direct, diffuse, and reflected light into electricity [2]. Bifacial panels also include performance parameters in their datasheets, but they also incorporate specific indicators related to their dual-sided operation.

One of the most relevant parameters is bifaciality, defined as the ratio of rear-side to front-side power output under standard test conditions (STC) [2]. From an analytical perspective, a bifacial module can be viewed as two surfaces contributing to energy production: the front side, which primarily receives direct irradiance, and the rear side, which captures reflected and diffuse radiation. However, bifaciality is determined by the module's design and cannot be modified. For this reason, the National Renewable Energy Laboratory (NREL) commonly emphasizes a more practical metric known as bifacial gain, which quantifies the additional energy produced by a bifacial module compared to its monofacial equivalent [7].

$$Bifacial\ Gain[\%] = \left(\frac{Ybifacial}{Ymonofacial} - 1\right) * 100 \quad (1)$$

The energy production of bifacial modules is affected by many environmental and installation factors. Among these, surface albedo plays a critical role, since highly reflective surfaces increase the amount of radiation reaching the rear side of the panel [8]. Installation height, module spacing, and structural alignment also affect the amount of reflected irradiance received [8]. In addition, bifacial systems remain subject to the same meteorological variables and atmospheric conditions that influence monofacial photovoltaic installations [9].

In order to integrate bifacial technology into modern electrical systems such as smart grids, predictive methods are needed to estimate both front and rear irradiance under real operating conditions. While numerous studies have successfully applied neural network models to forecast front-side irradiance, research on rear-side irradiance prediction in bifacial modules remains limited, despite its critical role in total energy production. This reveals the lack of precise models for forecasting next-day rear irradiance in bifacial panels using historical time-series and meteorological data.

Without detecting or leveraging correlations among these variables, patterns enabling reliable forecasts cannot be identified. This work addresses this gap by exploring predictive models that forecast the irradiance of the following day using historical time-series data and some environmental variables. The lack of accurate models for rear irradiance estimation limits the efficient integration of bifacial systems into smart grids. Improving the prediction of rear irradiance is therefore essential to exploit the advantages of bifacial technology and to support the reliability, stability, and sustainability of future smart grid infrastructures.

2.2 RELATED WORK

A review of the literature on photovoltaic energy forecasting reveals a wide range of machine learning approaches applied to this problem. For instance, Sabareesh et al. developed forecasting models using LSTM and GRU architectures to predict both weather conditions and system load in standalone systems, achieving a MAE of 18.71 and an R^2 of 0.98 for irradiance prediction [11]. More advanced approaches have focused on hybrid architectures. Bashir et al. introduced two novel models, CNN-ABiLSTM and CNN-Transformer-MLP, which combine CNNs with Attention-Based Bidirectional Long Short-Term Memory (ABiLSTM), Transformer-based structures, and Multi-Layer Perceptrons (MLP), respectively. In these models, CNN components capture short-term patterns, while sequential and attention-based mechanisms address long-term dependencies [4].

In a similar direction, Sharma et al. evaluated multiple hybrid architectures, including Random-Forest-BiLSTM (RF-BiLSTM), CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-Transformer, combining convolutional and recurrent components to extract spatial and temporal features from historical time series [12]. Additionally, Mariappan et al. implemented a CNN-LSTM model for forecasting global horizontal irradiance, incorporating metaheuristic algorithms to optimize hyperparameters [13]. Furthermore, Kumar Dhaked et al. conducted a comprehensive review of deep learning-based forecasting techniques, highlighting their strengths and limitations in solar energy applications [16].

However, when focusing specifically on bifacial photovoltaic systems, the available literature becomes significantly more limited. Sierra developed an LSTM-based model to estimate the energy production of bifacial modules, using the System Advisor Model (SAM) as a simulation tool for validation [3]. These findings indicate a clear research gap between the extensive work on monofacial photovoltaic forecasting and the limited number of studies addressing bifacial systems, especially in terms of rear irradiance prediction. While existing deep learning approaches provide a strong methodological foundation, their application to bifacial configurations using real-world data remains unexplored. This gap motivates the development of more robust and specialized models for accurate rear irradiance forecasting.

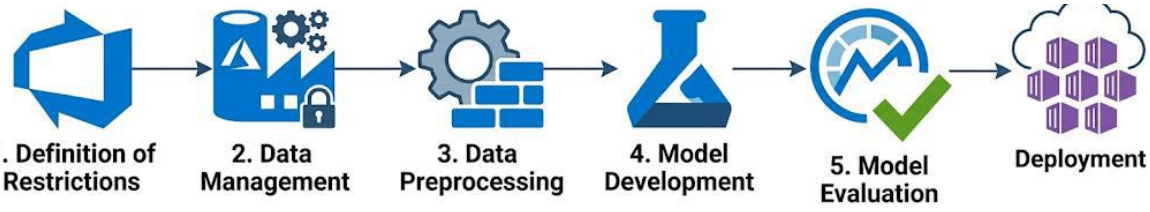
3. DEEP LEARNING FRAMEWORK FOR IRRADIANCE FORECASTING

Developing machine learning models involves a careful process at every stage. Before model implementation, several critical steps must be addressed, including data preprocessing, cleaning, and the proper handling of temporal dependencies. In addition, careful consideration is required to avoid issues such as data leakage and to ensure that the models capture meaningful patterns and generalize effectively. This chapter establishes the deep learning framework used for irradiance forecasting, outlining the methodological steps and design considerations required to build reliable time-series prediction models.

3.1 FORECASTING METHODOLOGY

The first step is to define the baseline framework, which establishes the scope and design criteria of the proposed methodology. The project's development is accomplished by following the stages shown in Figure 1.

Figure 1. Diagram showing the project flow in stages.



Note. Source: Generated using Google Gemini.

As shown in Figure 1, the methodological process was structured in five main phases. Initially, the restrictions were defined, encompassing data collection, defining the forecast time horizon, gap, size of the input time series window, number of forecast steps, and whether it was a direct or indirect forecast. Next, data management was carried out, covering the collection and selection of exogenous variables, strategies to prevent data leakage, data types and structure, and their characterization. Following this, data preprocessing was performed. This stage includes cleaning and handling null data, dividing the dataset into training, validation, and testing data, creating time windows, and transforming the data using scalers. Next came model development, where the architectures to be used were defined, models were created, hyperparameters were optimized, and these models were trained. Finally, the models were evaluated and deployed.

The project follows a sequential structure; however, it should be noted that in practice, the selection of exogenous variables and the characterization of the data are carried out after data cleaning and handling of null values, and not before. The following section defines the problem setting and the decisions associated with this first stage.

Forecasting tasks are typically categorized according to the forecasting time horizon, ranging from ultra-short term (up to 1 h), short-term (daily), medium-term (daily to weekly), and long-term (weekly to yearly) [4]. Each one fulfills a specific function in the context of smart grids, in addition to the special considerations that must be taken into account to make the forecast and train the models in each case.

Likewise, we also need to define how many steps ahead we want to predict [1], but to know that, we are missing a variable related to our data, and that is granularity, which in our time series context is known as data resolution. Typically, irradiance data is collected daily (Peak solar hours), hourly, or every 15 minutes, depending on the database or API used to retrieve

the data. For hourly data, an ultra-short-term forecast corresponds to a single step, whereas a 15-minute resolution leads to a multi-step forecast. In this last case, there are also two approaches; if you have a multi-step forecast, you can turn it into a single-step by predicting the value of the next step and using that prediction as input to predict the next one and so on [14]. However, this is not recommended for most problems and should be carefully considered before implementation [15].

Finally, the problem can be classified as direct or indirect prediction [1]; that is, the objective of the model is our variable of interest, or it is necessary to perform an additional calculation or take into account other variables.

Regarding the size of the input time series window, there is no universal rule that tells us how much data the model requires as input, considering the time resolution of the data and the forecast horizon. However, we can get an idea of the input window size based on the autocorrelation of the irradiance. A common rule of thumb is to use input windows that are roughly four times the forecast horizon. However, it must be taken into account that solar irradiance has a strong daily dependence but loses correlation as the forecast time horizon increases [16]. Table 1 shows a recommendation, but not a mandatory rule, on how to choose this window.

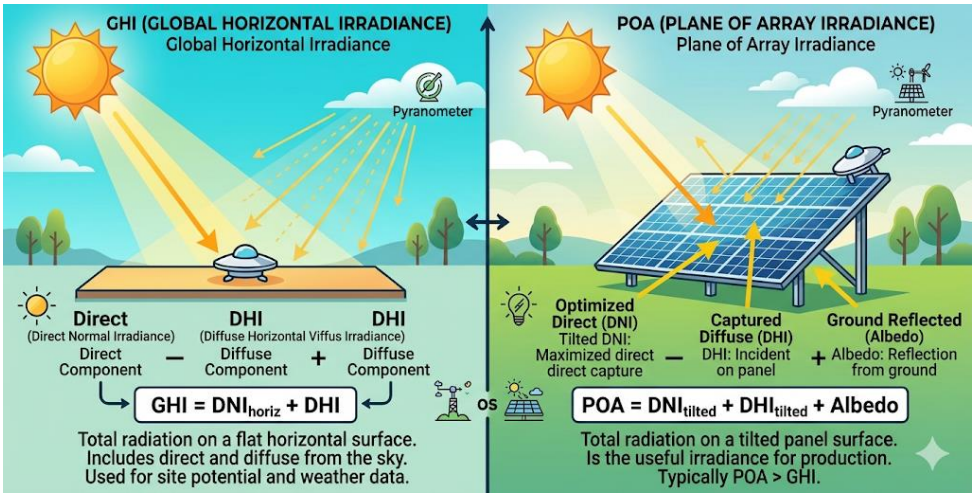
Table 1. Input data window recommendation based on time horizon.

Time Horizon	Window size (Hours)	Number of steps (15 Minutes)
1 hour	2-6	8-24
1 day	24-96	96-384
1 week	168-336	672-1344

It is also important to identify the origin of the data and how it was collected. When discussing irradiance, there are many types and methods for measuring it. For example, Power NASA uses three main types: Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI), and Diffuse Horizontal Irradiance (DHI), each with its own variations [17]. However, the irradiance most directly related to energy production is Plane of Array Irradiance (POA), as it measures the irradiance directly reaching the panel, as explained in the Figure 2.

Since irradiance is the main data point of our project, it is important to emphasize that the sensors used to measure it must be calibrated, and that communication and data collection problems must be avoided, as understanding how the sensor works, its characteristics and its operating modes facilitates data processing and subsequent analysis.

Figure 2. Difference between GHI and POA.



Note. Source: Generated using Google Gemini.

3.2 EXOGENOUS VARIABLES

The collection, analysis, and selection of exogenous variables can range from a simple process to a highly complex task, depending on the problem and research objectives. In irradiance forecasting, these variables play a critical role, as they provide additional information about the environmental conditions influencing solar radiation. A key aspect in selecting exogenous variables is their spatial resolution. Ideally, these variables should be collected at the same location as the target variable (irradiance). However, this is not always feasible, as it would require either proximity to a weather station or the installation of dedicated measurement systems for each photovoltaic setup.

As a result, external databases are commonly used, such as NASA POWER, ERA5, and NSRDB, as well as national meteorological sources. The following are some examples of spatial resolutions of well-known databases (1° is approximately 111 km).

- NASA Power: 1° X 1° for solar radiation and 0.5° X 0.625° for meteorological data
- ERA5: 0.1° X 0.1 ° (ERA5-Land) and 0.25° X 0.25° (ERA5) [18].

- NSRDB: 0.038° X 0.038° [19].

When selecting a data source, it is important to consider its spatial resolution, temporal consistency, and update frequency. For example, NSRDB offers high spatial resolution, while ERA5 and NASA POWER provide coarser resolutions but more frequent updates. Additionally, local topographic conditions must be considered, as regions with abrupt terrain variations may introduce significant discrepancies in meteorological variables.

Once the data has been collected, the next step is the selection of the most relevant exogenous variables. A commonly used baseline method is Pearson's correlation, which measures linear dependence between variables in the range [-1, 1]. Variables with higher absolute correlation values are typically selected as initial candidates. This approach is simple and interpretable, and it is often used as a preliminary filtering step [20].

More advanced methods of feature selection have also been proposed in the literature, but since they were not used in the present project, a description of these strategies has been left in Annex A. The selection of exogenous variables ultimately depends on the specific forecasting problem, the availability of data, and the desired balance between model complexity and interpretability.

3.3 DATA LEAKAGE PREVENTION STRATEGY

Ensuring a consistent and well-defined methodology for time series forecasting is insufficient if the resulting models are affected by data leakage. Data leakage occurs when a model is exposed, directly or indirectly, to information that would not be available at prediction time [23], leading to overly optimistic performance estimates and compromised generalization capability. Despite its critical impact, data leakage is frequently overlooked in the literature, potentially biasing conclusions and undermining the validity of reported results. For this reason, a rigorous treatment of this issue is essential.

Formally, data leakage is defined as the use of information during model training or validation that would not be accessible during the deployment or inference stage [23]. In this work, four primary sources of data leakage are identified.

- Overlap leakage: Occurs when test data is directly used for training or hyperparameter tuning [24].
- Multi-test leakage: Occurs when test data is used repeatedly for evaluating the model and making decisions such as algorithm selection, model selection, and hyperparameter tuning [24].
- Pre-processing leakage: Occurs when test data is merged with the training data for preprocessing. For example, feature selection and normalization [24].
- Target leakage: Models include data that will not be available when the model is used to make predictions [23].

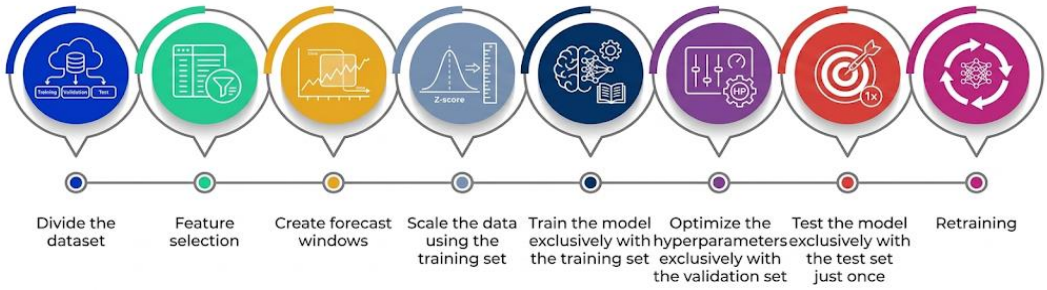
In practical terms, preprocessing leakage may occur when normalization is applied using the full dataset, or when independent scalers are fitted separately on training, validation, and test sets. Similarly, target leakage arises when future information is inadvertently incorporated into the input features. Overlap leakage can result from combining validation and test sets for hyperparameter tuning, while multi-test leakage emerges when repeated evaluation on the test set influences model selection.

Additionally, two special cases of data leakage are considered. The first involves generating time windows prior to splitting the dataset. In this scenario, leakage affects the input structure rather than the prediction target, meaning that future information is not explicitly used for forecasting [25]. While this approach is sometimes adopted to increase the amount of training data, it might still compromise the model's ability to generalize depending on the amount of repeated information in the context. The second case concerns feature selection performed on the entire dataset. Although this may have limited impact in large datasets with simple selection criteria, it can introduce significant bias when the selected features differ from those obtained using training data alone [26]. The presence of data leakage can lead to several critical issues, including inflated performance metrics, poor generalization to unseen data, biased model selection, and unreliable conclusions [23].

Following this principle, the proposed methodology enforces a strict separation between training, validation, and test sets. The intended pipeline applies feature selection, window generation, and scaling after temporal splitting; however, some RNNs models are discussed separately because their windows were generated before the split. The model is trained exclusively on the training set, hyperparameters are optimized using the validation set, and

final evaluation is conducted on the test set, which is used only once. Optionally, after hyperparameter tuning, the model may be retrained using the combined training and validation sets, with final performance assessed on the test set. While this approach can improve model robustness, it introduces a potential risk of soft overfitting. A flowchart representation of this workflow is provided in Figure 3.

Figure 3. Flowchart showing the steps to prevent data leakage.



Note. Source: Generated using Google Gemini.

3.4 DATASET

A comprehensive understanding of the dataset is essential for ensuring consistency, reliability, and reproducibility in time series forecasting tasks. Data characterization involves analyzing the structure, origin, and properties of the available data, and should be performed iteratively throughout the project, both before and after preprocessing. This process enables the identification of inconsistencies, potential errors, and relevant patterns that directly impact model performance. Since multiple data sources are involved, it is necessary to ensure a consistent time resolution across all variables. When discrepancies arise, alignment is achieved through resampling techniques. This step is critical to guarantee coherence between the target variable and the associated exogenous inputs.

The dataset used in this study integrates heterogeneous sources of information. Irradiance measurements are obtained from on-site sensors, providing high-resolution and reliable ground truth data. In contrast, meteorological variables are typically extracted from external databases, which may differ in resolution and accuracy. Additionally, temporal features are engineered using cyclic transformations (Sine and cosine encoding of daily and annual cycles), a widely adopted approach that enables machine learning models to capture periodic patterns inherent in solar irradiance data [27].

From a structural perspective, although datasets may initially contain heterogeneous data types (e.g., timestamps or categorical indicators), all inputs must ultimately be transformed into numerical representations to be processed by deep learning models. In this context, the dataset used in this study is primarily composed of continuous numerical variables, such as irradiance, temperature, along with engineered features derived from temporal information.

Ensuring a consistent numerical representation is essential, particularly for time-related variables, which must be properly encoded to preserve temporal dependencies. Deep learning models operate on tensor structures, where input data is represented as multi-dimensional arrays of floating-point values. In addition to numerical consistency, structural coherence of the data must be guaranteed. This includes maintaining consistent dimensionality, alignment across variables, and compatibility between input and output shapes when performing operations such as windowing, batching and model training.

Beyond data representation, it is also necessary to characterize the dataset in terms of its size, temporal coverage, and completeness. This includes quantifying the number of observations, analyzing data distribution over time, and identifying missing or anomalous values [28]. Such analysis is critical in deep learning models, as model performance is highly dependent on data quality, consistency, and the representativeness of training samples.

3.5 DATA PREPROCESSING PROTOCOL

Raw data cannot be directly used for training deep learning models; therefore, a structured preprocessing pipeline is required to ensure data quality, consistency, and suitability for time series forecasting. The proposed protocol consists of the following stages.

1. **Data cleaning**: The first step involves removing irrelevant or potentially harmful information from the dataset. This includes eliminating duplicate columns, correcting inconsistent data types, and discarding variables that do not contribute to the predictive task [29]. This process is performed after initial data characterization and ensures that only meaningful features are retained.

2. **Temporal consistency adjustments**: Time-related inconsistencies must be carefully addressed. In this project, the dataset includes measurements affected by daylight saving time (DST), which introduces artificial temporal shifts due to clock

adjustments rather than physical changes in the system. Since this behavior does not reflect real irradiance dynamics, a correction is applied prior to merging all data sources. Additionally, data collected before a specific date was removed due to identified issues in measurement devices, ensuring the reliability of the final dataset.

3. **Handling null data**: Handling missing values is particularly critical in irradiance time series. In some cases, null values may correspond to nighttime periods, where irradiance should be zero rather than missing. Therefore, distinguishing between true missing data and physically meaningful zero values is essential. The treatment of missing data depends on the forecasting objective. When predicting full-day profiles, even a single missing value within the day may compromise the entire sample. In such cases, affected periods are either removed or carefully imputed, ensuring that no artificial information or data leakage is introduced. Furthermore, missing values can disrupt temporal continuity. Since time series models rely on sequential input windows, gaps in the data must be explicitly considered during window generation. Failure to do so may result in unrealistic input-output relationships.

4. **Dataset splitting**: The dataset is divided into training, validation, and test sets following a strictly temporal approach. Random splitting is avoided, as it would break temporal dependencies and introduce data leakage. The training set is used for model fitting, the validation set for hyperparameter tuning, and the test set for final evaluation [30]. Typical splitting ratios used in this work are summarized in Table 2.

Table 2. Recommended splitting for time series models.

Train	Validation	Testing	Features
80%	10%	10%	Large datasets, intensive tuning
75%	10%	15%	Balanced, representative test
70%	15%	15%	Medium-sized datasets

5. **Time window generation**: Input-output sequences are constructed using sliding windows based on the defined forecasting horizon and input length. This process transforms the dataset into a supervised learning format suitable for deep learning models. Special care is taken to ensure that windows do not cross discontinuities caused by missing data. Only temporally consistent sequences are considered, preserving the chronological integrity of the data.

6. **Data scaling and normalization**: Feature scaling is applied to ensure numerical stability during model training. All transformations are fitted exclusively on the training set to prevent data leakage. Three main scaling approaches are considered:

    - **MinMax scaling**: Maps data to a fixed range, typically [0, 1], but is sensitive to outliers [31].
    - **Standard scaling**: Centers data using mean and standard deviation, suitable for normally distributed variables [31].
    - **Robust scaling:** Uses median and interquartile range, providing robustness against outliers [31].

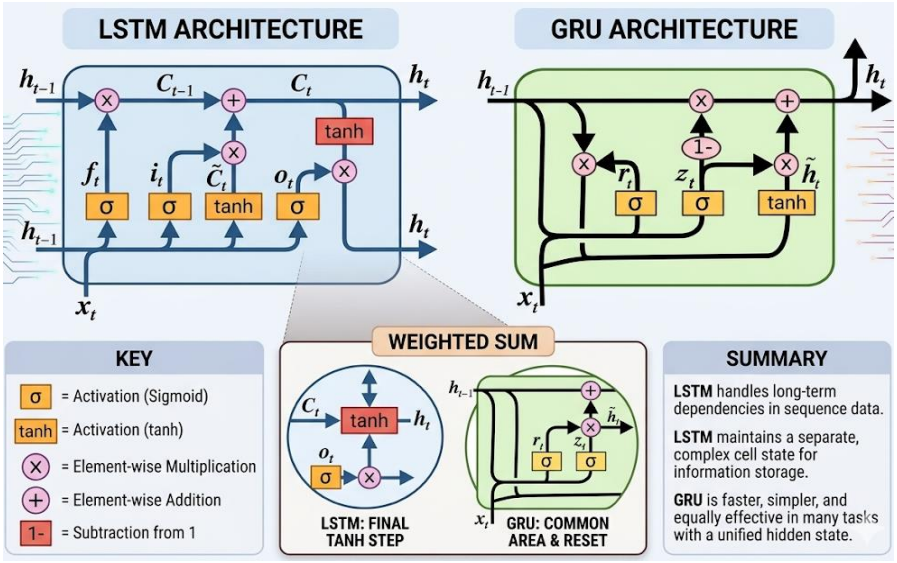
    Finally, the selected transformations are consistently applied to the training, validation, and test sets. Each variable is scaled according to its characteristics, ensuring a coherent and physically meaningful representation of the data.

3.6 PREDICTIVE MODELS BASED ON TIME SERIES DATA

Deep learning models are widely used for time series forecasting due to their ability to capture nonlinear patterns and temporal dependencies [1]. In this work, several neural network architectures are evaluated to model irradiance dynamics. Feedforward Neural Networks (FFNN) represent the simplest architecture, where information flows sequentially through fully connected layers [16]. However, their ability to model temporal dependencies is limited, making them less suitable for time series problems. RNNs are specifically designed to handle sequential data by incorporating information from previous time steps through internal states. Nevertheless, standard RNNs often suffer from vanishing and exploding gradient problems, which hinder their ability to learn long-term dependencies [1].

To address these limitations, more advanced architectures such as LSTM and GRU are employed. These models introduce gating mechanisms that regulate the flow of information, allowing them to retain relevant temporal features over longer sequences. While LSTM networks use separate input, output, and forget gates, GRU simplifies this structure by combining certain gates, resulting in lower computational complexity [1]. Figure 4 shows a representation of an LSTM and a GRU cell.

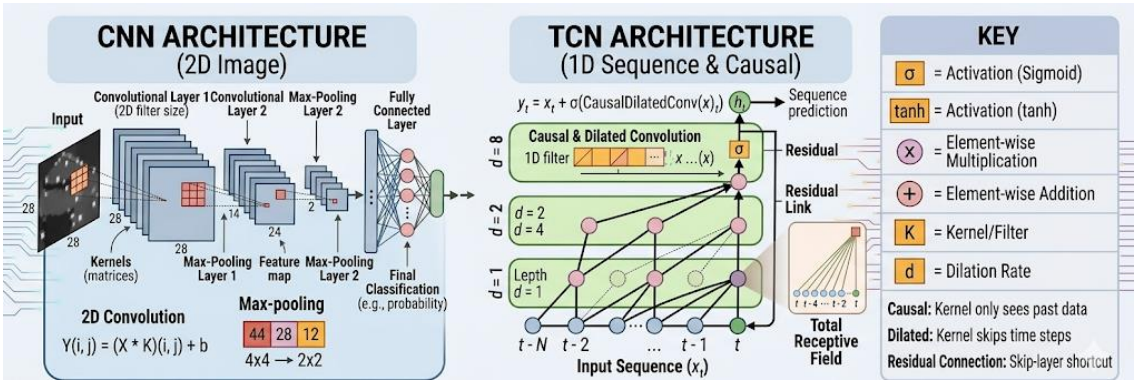
Figure 4. Representation of an LSTM cell and a GRU cell.



Note. Source: Generated using Google Gemini.

Bidirectional variants, such as BiLSTM and BiGRU, extend these architectures by processing sequences in both forward and backward directions [1]. This enables the model to capture additional contextual information, improving performance in scenarios where both past and future dependencies are relevant. In addition to RNNs, convolutional approaches are also considered. One-dimensional Convolutional Neural Networks (1D-CNN) can extract local patterns from time series data [16], while TCNs introduce causal and dilated convolutions to model long-range dependencies efficiently [32]. These architectures provide an alternative to recurrent models with competitive performance.

Figure 5. Diagram of CNNs and TCNs.



Note. Source: Generated using Google Gemini.

Hybrid models combine the strengths of different architectures. In this work, CNN-LSTM and CNN-GRU models are implemented, where convolutional layers first extract local features,

followed by recurrent layers that capture temporal dependencies [10]. These combinations often enhance predictive performance by leveraging both spatial and temporal representations. The design of each model depends on the forecasting objective, input structure, and selected variables. All architectures follow a general structure consisting of an input layer, one or more feature extraction layers (recurrent or convolutional), and fully connected layers for final prediction.

For sequential models (LSTM, GRU, BiLSTM, BiGRU), two stacked layers are typically used to improve representation capacity, with the final layer configured to return a single output sequence. In hybrid models, a convolutional block is followed by a single recurrent layer.
Activation functions are selected according to the role of each layer. Recurrent layers commonly use hyperbolic tangent functions, while convolutional and dense layers employ ReLU activation [1]. The output layer uses a linear activation function. Typical activation functions used in this work are summarized in Annex B. To reduce overfitting, dropout layers are incorporated between hidden layers, typically within a range of 0.1 to 0.3. Model training is performed using gradient-based optimization methods. The objective is to minimize a loss function by iteratively updating model parameters. Among the available optimizers, adaptive methods such as Adam are commonly preferred due to their efficiency and stability [33].

Training is conducted over multiple epochs, balancing convergence speed and generalization performance. Additionally, callbacks such as early stopping are employed to prevent overfitting by halting training when validation performance no longer improves.
Model performance is highly dependent on the selection of hyperparameters, including the number of units, kernel sizes, dropout rates, and learning rates. To identify suitable configurations, automated hyperparameter tuning is performed using methods such as Random Search and Bayesian Optimization. The choice of optimization strategy depends on the size of the search space and computational constraints. While Random Search is effective for smaller spaces, more advanced approaches such as Bayesian Optimization provide a more efficient exploration of complex hyperparameter configurations [34].

3.7 EVALUATION METHODOLOGY

In the previous chapter, the concept of a loss function was introduced in the context of model training. This section focuses on the evaluation metrics and methodologies used to assess model performance and interpret the obtained results.

The loss function provides a quantitative measure of model error during training and is optimized iteratively across epochs. Its selection must be aligned with the nature of the problem, as it directly influences model behavior [16]. In regression tasks, common choices include metrics that are minimized, such as MAE. Selecting an inappropriate loss function may introduce bias and lead to misleading results. In this work, MAE is selected as the loss function due to its stability across the full irradiance range, including both daytime and nighttime conditions, and its ability to provide a balanced representation of prediction errors without overemphasizing extreme values. To enable a more comprehensive evaluation, the metrics MAE, R^2, and MAPE [1] are considered and used to assess model performance from different perspectives. Their formulations are given by:

$$MAE = \frac{1}{n} * \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (2)$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \quad (3)$$

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

It is important to note that MAPE presents limitations in irradiance forecasting, particularly for values close to zero, where the metric becomes unstable. For this reason, MAPE must be interpreted with caution, especially when near-zero irradiance values are included.

A key challenge in model evaluation arises from the temporal structure of the data. Since the forecasting horizon corresponds to one day ahead, evaluating the entire test set as a single error would obscure relevant temporal variability. Therefore, performance is assessed by treating each day as an independent evaluation unit. However, this approach introduces additional complexity. Days with low irradiance often lead to distorted metric values, such as negative R^2 or low MAPE values, making direct averaging across days unreliable.

Consequently, a robust evaluation strategy is required to ensure meaningful comparisons. One of them is that if you have high-resolution meteorological data, you can classify the day based on that data. Since the resolution is high, the approximation of what happens at the photovoltaic installation is very good, and the day's classification can serve as a good indicator with a sufficiently high temporal and spatial resolution. But this project doesn't have sufficient resolution to make this approach feasible, so a statistical methodology is proposed to classify days by their level of difficulty. This approach is based on the following steps:

1. Divide the models by sensor.
2. Select a daily metric for each model and sensor.
3. For each day, take the median of that metric across all models. The median is robust to outliers, so the final result is a neutral value.
4. Use the median to classify the days by difficulty and by quartiles.

This methodology offers advantages such as:

- All models will be evaluated in the same quartiles.
- The ranking depends on all the models created.
- It reduces reliance on visual criteria and provides a more reproducible ranking.

4. IMPLEMENTATION, EXPERIMENTS AND RESULTS

This chapter addresses the application of a real-world case study, where the methodology described in the previous chapter is applied. Therefore, this chapter describes the specific case study, some of the decisions made, the results obtained, and some graphs that will aid in the model ablation and results analysis of the project.

4.1 CASE STUDY DESCRIPTION

To begin describing the case study, we first define the main problem and its constraints. The data were obtained from the University of Delaware's (UD) Institute of Energy Conversion (IEC) from July 2021 to May 2025, where several bifacial modules were adapted for the research. These modules collected environmental, electrical system, albedo, and power optimizer data with a 15-minute resolution. Our variables of interest are the irradiance measured on the front and rear surfaces and the albedo. In this case, two meters were used on the front side to measure the POA irradiance; meanwhile, on the rear side, three meters were used to also measure the exact irradiance reaching that side.

Although the rear irradiance is not constant across its surface, the IEC typically uses the average as an approximation of the average rear irradiance. As mentioned earlier, our goal is to generate 24-hour forecasts, so an input period of 4 days, or 384 fifteen-minute data points, was chosen. This study focuses on direct multi-step irradiance forecasting; extensions toward power prediction fall outside the present scope. Additionally,

A zero-gap setting was adopted because the predictor variables used at inference time are assumed to be available up to the forecast origin.

4.2 IMPLEMENTATION OF THE PROPOSED METHODOLOGY

This section describes the practical implementation of the proposed methodology, including data integration, preprocessing, feature selection, and model configuration. To enhance model performance, some exogenous variables were incorporated from external sources. Meteorological data were obtained from NASA POWER and ERA5. From NASA POWER, temperature (temp), precipitation (precip), and both relative (%RH) and specific humidity (q) were selected, while total cloud cover (tcc) was obtained from ERA5. Given the uniform topography of the study area, these variables provide a reasonable approximation of local environmental conditions. Due to differences in temporal resolution, alignment strategies were applied. Hourly NASA POWER data were sampled by value replication to match the 15-minute resolution, while ERA5 variables were interpolated using a smoothing approach.

The initial dataset consisted of 135,401 observations. After incorporating missing albedo information and restricting the data to a reliable temporal range, additional temporal features (daily and yearly cyclic encodings) were introduced. Following data cleaning and preprocessing, the dataset was reduced to 132,768 observations (1,383 days). After applying missing data filtering criteria, the final dataset contained 125,664 observations (1,309 days). Missing values introduced discontinuities in the time series, which were explicitly identified and considered during subsequent window generation to preserve temporal consistency. A summary of the dataset distribution is presented in Table 3.

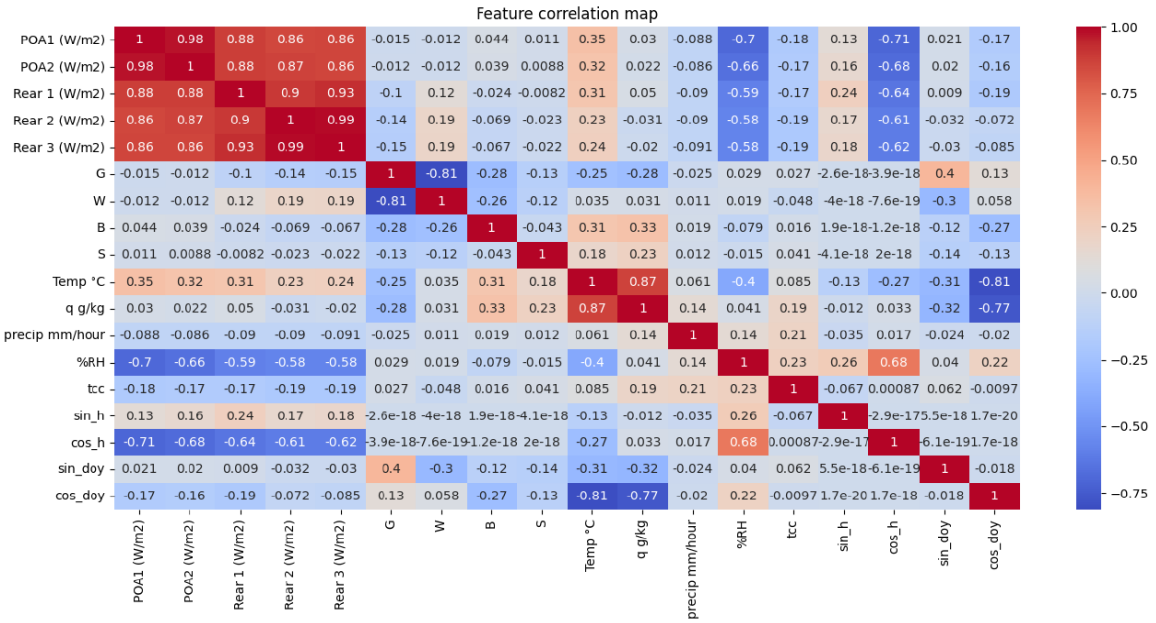
Table 3. Distribution of available data points by surface type and year.

	Gravel	White	Black	Silver	2021	2022	2023	2024	2025
Data points	47232	47040	29376	2016	14880	34848	34848	28512	12576

Feature relevance was assessed using Pearson correlation analysis, computed on the training dataset to avoid potential bias. The resulting correlation matrix (Figure 6) highlights strong relationships between irradiance and variables such as temperature and relative humidity. Based on this analysis, temp and %RH were included in all models, while

additional variables such as tcc, precip, and q were incorporated progressively to evaluate their impact. Albedo was specifically included in rear irradiance and integrated models due to its physical relevance. Furthermore, the observed correlation among irradiance measurements from different sensors motivated the exploration of integrated models, where multiple irradiance signals are used jointly as inputs.

Figure 6. Feature correlation map



Different modeling approaches were implemented with varying levels of data leakage control. In recurrent models (LSTM, GRU, BiLSTM, BiGRU), input windows were generated before dataset splitting, introducing minor contextual overlap, so Group 1 should be interpreted as having a weaker leakage-control scheme than Group 2. Although this does not involve direct use of future target values, it may reduce generalization capability.

In contrast, convolutional and hybrid models (TCN, CNN-LSTM, CNN-GRU) strictly followed the proposed preprocessing pipeline, ensuring complete separation between training, validation, and test data. These models (Group 2) are therefore treated as free of the leakage sources explicitly controlled in the proposed pipeline. The dataset was split using an 80/10/10 temporal partition for training, validation, and testing. Input-output sequences were generated considering the defined forecasting horizon, input window size, and identified temporal gaps. Special care was taken to avoid constructing windows across discontinuities introduced by missing data. Both single-sensor and integrated model configurations were considered in this project.

To evaluate the impact of preprocessing choices, two scaling strategies were tested: MinMax scaling and Robust scaling. These were applied consistently across all model configurations. All models were trained using early stopping (patience = 8) and model checkpointing to ensure optimal performance. Hyperparameter optimization was performed using Bayesian optimization, and the best configurations were subsequently used to retrain models on extended datasets.

In total, 560 individual models and 112 integrated models were developed, representing a comprehensive exploration of deep learning approaches for front and rear irradiance forecasting. Detailed configurations and hyperparameter settings are provided in the Annex C. After training and storing all model outputs, a post-processing pipeline was developed to systematically organize and analyze the results.

Based on this structured data, a set of functions was implemented to compute daily median error values, following the evaluation methodology described in the previous section. This allowed each day to be classified into difficulty quartiles for both groups under analysis, and the results were subsequently consolidated into a single tabular format for further interpretation. Initial classification was performed using the median MAE as the reference metric. However, this approach proved insufficient, as the resulting R^2 and MAPE values remained inconsistent even within intermediate quartiles, limiting the interpretability of model performance across different conditions. To address this limitation, the classification procedure was revised by replacing MAE with the median R^2 as the reference metric. This modification led to a significant improvement in the consistency and interpretability of the results. In particular, R^2 demonstrated a greater sensitivity to low-irradiance conditions, enabling a more accurate identification of challenging forecasting scenarios.

Furthermore, the quartile-based classification showed strong agreement between the evaluated groups, with both approaches consistently identifying the same days as easy or difficult. This indicates that R^2 provides a more robust and discriminative basis for performance-based categorization. These findings suggest that, in the context of irradiance forecasting, applying the quartile-classification procedure, R^2 provided a more discriminative ranking than MAE under low-signal conditions. Consequently, this metric is adopted as the primary reference for subsequent analysis, serving as the foundation for a more detailed evaluation.

4.3 FIRST MACHINE LEARNING GROUP RESULTS

Once the quartile-based classification using the R^2 metric was obtained, all models were organized according to architecture, scaler, model type, number of exogenous variables, retraining strategy, and quartile. This structured organization enabled a consistent comparison across configurations and facilitated the identification of general performance patterns. It is important to note that quartiles are defined cumulatively: Q25 includes only the easiest days, Q50 includes easy and intermediate days, Q75 extends to moderately difficult conditions, and Q100 represents the complete test set.

To support the visualization and exploration of results, all metrics were integrated into Power BI Desktop, from which the figures presented in this chapter were generated.
A first observation across all configurations is related to model stability, assessed through the standard deviation of the R^2 metric. For easy (Q25), intermediate (Q50), and moderately difficult conditions (Q75), BiGRU consistently exhibited the lowest variability, indicating a stable performance across different scenarios. In contrast, BiLSTM showed competitive stability in specific cases, particularly for rear sensors under selected quartiles.

Table 4. Representative best-performing configurations for Group 1.

Sensor	**Architecture**	**Suggested optimal configuration**
POA1 / POA2	BiGRU / GRU	Individual, MinMax Scaler, No Retraining.
Rear 1	BiLSTM / GRU	Individual, MinMax (Q50) or Robust (Q75), and Retraining.
Rear 2	LSTM / BiGRU	Integrated, Robust Scaler, No Retraining.
Rear 3	BiGRU / GRU	Integrated, MinMax Scaler, No Retraining.

To provide a concise overview of performance tendencies, Table 4 summarizes the most competitive configurations identified for each sensor, based on aggregated results across quartiles and evaluation metrics. Finally, Figure 7 presents an example of the interactive

report developed for the POA1 sensor, which integrates the evaluated configurations and corresponding performance metrics.

Figure 7. Overall results for model ablation for the POA1 sensor.

4.4 SECOND MACHINE LEARNING GROUP RESULTS

The same classification framework was applied to the second group of models, enabling a consistent comparison across architectures, configurations, and quartiles. A primary observation relates to model stability, again evaluated through the standard deviation of the R^2 metric. For easy conditions (Q25), CNN-GRU exhibits the lowest variability for POA1, POA2, and Rear 2 sensors, while CNN-LSTM shows greater stability for Rear 1 and Rear 3. As the level of difficulty increases (Q50), CNN-GRU maintains stable performance for POA sensors and Rear 2, whereas CNN-LSTM remains competitive for Rear 1. In contrast, for more challenging scenarios (Q75), TCN demonstrates the lowest dispersion in most cases, indicating a more consistent behavior under increased variability, although not necessarily superior predictive accuracy.

To summarize the observed performance tendencies, Table 5 presents representative configurations for each sensor, based on aggregated results across models and quartiles.

Table 5. Representative best-performing configurations for Group 2.

Sensor	Architecture	Suggested optimal configuration	Observations
POA1	**CNN-GRU / CNN-LSTM**	Individual, Robust Scaler, No Retraining.	CNN-GRU leads on average; CNN-LSTM is better in Q75.
POA2	**CNN-GRU / CNN-LSTM**	Individual (Q75), Robust Scaler, No Retraining.	Integrated models are only competitive in the Q50.
Rear 1	**CNN-GRU / CNN-LSTM**	Integrated, Robust Scaler, No Retraining.	CNN-GRU dominates the average; CNN-LSTM is better in MAPE for Q75.
Rear 2	**CNN-LSTM / CNN-GRU**	Integrated (Q50) / Individual (Q75), Robust Scaler.	In most cases, CNN-LSTM is superior in MAE and R^2.
Rear 3	**CNN-LSTM / CNN-GRU**	Integrated (Q50) / Individual (Q75), Robust Scaler.	CNN-LSTM leads in MAE/MAPE; CNN-GRU better captures R^2.

Figure 8. Overall results for model ablation for the Rear 2 sensor.

Finally, Figure 8 presents an example of the interactive report developed for the Rear 2 sensor, integrating the evaluated configurations and corresponding performance metrics. This visualization complements the comparative analysis and supports the discussion presented in the following chapter.

4.5 RESULTS ANALYSIS AND MODEL ABLATION

This section presents a unified analysis of the obtained results, integrating ablation studies, model comparisons, and performance interpretation across different configurations. The discussion focuses on identifying consistent patterns across architectures, scaling strategies, model structures, and exogenous variables, rather than isolated results per sensor or quartile. Detailed results by quartile and number of exogenous variables are provided in Annex D.

1. **Architecture performance and stability:** Across both groups, recurrent and hybrid architectures generally outperformed simpler configurations across most sensors and quartiles. For Group 1, GRU and BiGRU models exhibited the most robust performance across all sensors and quartiles, particularly in terms of stability and average accuracy. Although LSTM and BiLSTM showed competitive results in specific scenarios, their performance was less consistent.

   In Group 2, CNN-based hybrid models (CNN-GRU and CNN-LSTM) outperformed TCN in most cases, especially for easy and intermediate conditions. However, TCN demonstrated superior stability under more challenging scenarios (Q75), suggesting a greater robustness to variability rather than higher predictive accuracy. Overall, these results indicate that hybrid architectures are more effective at capturing complex spatiotemporal relationships, while GRU-based models provide a strong balance between accuracy and stability.

2. **Individual and Integrated models**: A clear distinction emerges between front and rear irradiance modeling. For front sensors (POA1 and POA2), individual models consistently achieved higher R^2 values, indicating that independent learning is sufficient to capture the dominant irradiance patterns. In contrast, for rear sensors, integrated models often showed improved performance, particularly when combined with appropriate scaling techniques. This suggests that rear irradiance benefits from

shared information across sensors, likely due to its higher variability and dependence on environmental interactions such as albedo and reflections.

3. **Effect of scaling methods**: The choice of scaler plays a critical role and varies depending on the sensor type and model architecture. For Group 1, MinMax scaling consistently produced better results for front sensors under low and moderate difficulty conditions (Q25 and Q50).

   However, as the level of difficulty increased, Robust scaling became more effective, particularly for rear sensors. In contrast, Group 2 models showed a stronger preference for Robust scaling across most scenarios, especially for intermediate and difficult conditions. This suggests that hybrid architectures are more sensitive to outliers and benefit from scaling techniques that reduce their influence.

4. **Impact of exogenous variables:** The inclusion of exogenous variables does not lead to uniform improvements and depends strongly on the model configuration. For front sensors, performance generally improves with the inclusion of additional variables, particularly when using MinMax scaling. However, diminishing returns are observed beyond three to five variables, depending on the architecture.

   For rear sensors, the impact is more complex. Rear 1 and Rear 2 benefit significantly from additional variables, especially under intermediate and difficult conditions, indicating a higher dependency on environmental factors. In contrast, Rear 3 shows early saturation, with no significant improvements beyond two variables. Additionally, some configurations show performance degradation when combining Robust scaling with excessive exogenous inputs, highlighting the importance of balanced feature selection.

5. **Effect of Retraining:** The impact of retraining is generally limited and, in most cases, does not lead to improved performance. In most evaluated configurations, models without retraining achieved higher R^2 values. Retraining only proved beneficial in specific scenarios, particularly for rear sensors under certain configurations. This behavior suggests the presence of mild overfitting when retraining is applied, reducing the model's ability to generalize to unseen data.

All models exhibited reduced performance under highly difficult conditions (Q100), confirming the intrinsic complexity of irradiance forecasting in low-signal scenarios. Despite this limitation, GRU/BiGRU (Group 1) and CNN-based models (Group 2) maintained relatively better performance in terms of MAE.

The results demonstrate that model performance is highly dependent on the interaction between architecture, scaling strategy, and input variables. No single configuration is universally optimal; instead, performance must be evaluated within the specific context of sensor type and forecasting difficulty. These findings provide a comprehensive understanding of the behavior of deep learning models for bifacial irradiance forecasting and establish a solid foundation for the conclusions presented in the next chapter.

From an explainability perspective, the observed results are consistent with the features of the evaluated architectures and the nature of the irradiance forecasting problem. RNN-based models such as GRU and BiGRU demonstrate strong performance due to their ability to capture temporal dependencies efficiently while maintaining a balance between model complexity and generalization, which explains their stability across multiple quartiles. Similarly, hybrid architectures (CNN-LSTM and CNN-GRU) outperform standalone models in several scenarios because they combine local feature extraction with temporal learning, making them particularly effective when exogenous variables are incorporated.

The impact of scaling techniques also aligns with expected behavior. The MinMax scaler enhances performance in stable conditions (Q25–Q50) by preserving relative variations, while the Robust scaler becomes more effective under higher variability (Q75), where outliers and noise are more prevalent, particularly in rear-side irradiance. The limited benefit of retraining in most scenarios suggests that the models already capture the dominant temporal patterns during initial training, and that additional retraining may introduce unnecessary variance rather than improving generalization.

Furthermore, the effect of exogenous variables reflects the physical characteristics of the system. Front-side irradiance, being more directly driven by atmospheric conditions, benefits from additional meteorological inputs, while rear-side sensors—especially Rear 3—exhibit saturation behavior, indicating that their variability is governed more by local structural or reflective conditions than by large-scale atmospheric variables.

5. CONCLUSIONS AND FUTURE WORK

This chapter summarizes the main findings of this research on deep learning models for irradiance forecasting. The study evaluated multiple architectures, preprocessing strategies, and the integration of exogenous variables under different levels of forecasting difficulty. The conclusions presented here synthesize the most relevant patterns identified throughout the experimental analysis and highlight the practical implications of the proposed methodology.

5.1 CONCLUSIONS

The results confirm that model performance strongly depends on the interaction between architecture, preprocessing techniques, and data characteristics. Within this case study, recurrent architectures—particularly GRU and BiGRU—provided the best balance between accuracy and stability in low and medium difficulty scenarios (Q25–Q50), making them suitable for relatively stable forecasting conditions. In contrast, hybrid models such as CNN-GRU and CNN-LSTM demonstrate superior robustness in more complex scenarios (Q75–Q100), benefiting from their ability to combine feature extraction with temporal learning.

Preprocessing plays a critical role in model performance. The MinMax scaler is more effective for front-side irradiance under stable conditions, while the Robust scaler improves performance in more variable scenarios, especially for rear-side sensors. Additionally, the results show that retraining is not generally beneficial, as it introduces variability without significantly improving generalization. The incorporation of exogenous variables enhances performance in front sensors, where irradiance is strongly driven by atmospheric conditions. However, rear sensors—particularly Rear 3—show saturation behavior, indicating that additional variables do not necessarily translate into better predictions. This highlights the importance of understanding the physical system when designing input features.

From a methodological perspective, the proposed evaluation framework based on quartile classification and median-based metrics proved to be effective for assessing model performance under different levels of difficulty. The use of R^2 as a classification metric enabled a more consistent identification of challenging days, improving the reliability of the analysis. Overall, the study demonstrates that there is no single optimal model. Instead, model selection should be guided by the level of variability in the data and the specific characteristics of the forecasting problem. In the evaluated case study, recurrent models

were preferable under more stable conditions, whereas hybrid architectures were more competitive in more complex scenarios.

5.2 FUTURE WORK

Future research could focus on a more advanced selection of exogenous variables, moving beyond linear approaches such as Pearson correlation toward methods capable of capturing nonlinear relationships, including feature importance techniques, mutual information, or model-based selection strategies. This would allow a more accurate representation of the underlying physical processes influencing irradiance.

Additionally, exploring different input window sizes and forecasting horizons could provide further insights into the temporal dynamics of the problem. Alternative configurations, such as shorter horizons or multi-scale approaches, may improve model adaptability and performance under varying operational conditions.

From an application perspective, future work may also address the integration of these forecasting models into smart grid environments, particularly in the context of distributed generation. In this regard, improving short-term forecasting reliability could contribute to better energy management, grid stability, and decision-making processes in systems with high penetration of renewable energy sources.

BIBLIOGRAPHY

[1] Kumar Dhaked, D., Narayanan, V. L., Gopal, R., Sharma, O., Bhattarai, S., & Dwivedy, S. K. (2025). Exploring deep learning methods for solar photovoltaic power output forecasting: A review. *Renewable Energy Focus*, *53*(100682), 100682. https://doi.org/10.1016/j.ref.2025.100682

[2] Burns & McDonnell. (2020). *Bifacial modules: 2 sides to every solar panel* [White paper / Bifacial solar panels]. https://www.firstsolar.com/-/media/First-Solar/Technical-Documents/Bifacial-Documents/Bifacial-Modules-2-Sides-To-Every-Solar-Panel-white-paper-burns-mcdonnell-09120.ashx

[3] Sierra, M. (2023). *Inteligencia artificial y paneles solares bifaciales para la optimización del autoconsumo energético* [Artificial intelligence and bifacial solar panels for optimizing energy self-consumption]. https://ainves.org/an5202389101

[4] Bashir, T., Wang, H., Tahir, M., & Zhang, Y. (2025). Wind and solar power forecasting based on hybrid CNN-ABiLSTM, CNN-transformer-MLP models. *Renewable Energy*, *239*(122055), 122055. https://doi.org/10.1016/j.renene.2024.122055

[5] Shawkat Ali, M. (Ed.). (2013). *Smart grids: Opportunities, developments, and trends* (2013a ed.). Springer.

[6] Rodríguez, L. (2021, febrero 26). *Bifacial modules: a comprehensive guide on financial and technical performance of the next hot thing in solar*. https://ratedpower.com/blog/bifacial-modules/

[7] Ovaitt, S. (2022). Understanding the bifacial modeling & field research space. Advance in Photovoltaics: Tandems & Bifacial PV. https://docs.nlr.gov/docs/fy23osti/84763.pdf

[8] Kathuria, H., Singh, I., Gupta, A., Puniya, G., & Kumar, B. (2024). Analysis of bifacial photovoltaic panel under different reflective surfaces. *2024 IEEE Third International Conference on Power Electronics, Intelligent Control and Energy Systems*, 933–938.

[9] Li, X., Zhang, X., & Cai, H. H. (2026). Impact of environmental factors on renewable energy generation in China: A scenario analysis using predictive modeling for solar energy systems. *Renewable and Sustainable Energy Reviews*, *234*(116807), 116807. https://doi.org/10.1016/j.rser.2026.116807

[10] Fathi, M., Allahyari, S., Ahmadi, A., Zahedi, A., & Asiaei, S. (2025). Enhanced global solar irradiance and temperature forecasting for rooftop PV power generation prediction using multi-step LSTM and GRU based hybrid models. *Energy Conversion and Management: X, Volume* (28). https://doi.org/10.1016/j.ecmx.2025.101351

[11] Sabareesh, S. U., Aravind, K. S. N., Chowdary, K. B., Syama, S., & Devi V S, K. (2023). LSTM based 24 hours ahead forecasting of solar PV system for standalone household system. *Procedia Computer Science*, *218*, 1304–1313. https://doi.org/10.1016/j.procs.2023.01.109

[12] Sharma, V., Panda, M. R., & Kar, B. (2026). Solar power forecasting using hybrid deep learning: Performance enhancement with random forest-BiLSTM and ensemble modeling. *Journal of Visualized Experiments: JoVE*, *228*. https://doi.org/10.3791/69743

[13] Mariappan, Y., Ramasamy, K., & Velusamy, D. (2025). An optimized deep learning based hybrid model for prediction of daily average global solar irradiance using CNN-LSTM architecture. *Scientific Reports*, *15*(1), 10761. https://doi.org/10.1038/s41598-025-95118-3

[14] Limouni, T., Yaagoubi, R., Bouziane, K., Guissi, K., & Baali, E. H. (2023). Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model. *Renewable Energy*, *205*, 1010–1024. https://doi.org/10.1016/j.renene.2023.01.118

[15] Limouni, T., Yaagoubi, R., Bouziane, K., Guissi, K., & Baali, E. H. (2022). Univariate and multivariate LSTM models for one step and multistep PV power forecasting. *International Journal of Renewable Energy Development*, *11*(3), 815–828. https://doi.org/10.14710/ijred.2022.43953

[16] Kumar Singh, P., Prakash, A., Saraswat, A., Gupta, Y., & Sabhahit, J. N. (2025). A compound of deep-learning and feature selection for solar power forecasting applications. *IEEE Access: Practical Innovations, Open Solutions*, *13*, 164742–164770. https://doi.org/10.1109/access.2025.3610419

[17] National Aeronautics and Space Administration (NASA). (2023-2025). *Prediction of Worldwide Energy Resources (POWER) [Base de datos]*. Langley Research Center. https://power.larc.nasa.gov/

[18] *ERA5 atmospheric reanalysis | Climate Data Guide*. (2025, 11 diciembre). NCAR. https://climatedataguide.ucar.edu/climate-data/era5-atmospheric-reanalysis#:~:text=ERA5%2C%20sucesor%20de%20ERA%2DInterim,hPa%20(aproximadamente%2080%20km).

[19] Sengupta, M., Xie, Y., Lopez, A., Habte, A., Maclaurin, G., & Shelby, J. (2018). *The national solar radiation data base (NSRDB)* Elsevier BV. https://doi.org/10.1016/j.rser.2018.03.003

[20] Jebli, I., Belouadha, F., Kabbaj, M. I., & Tilioua, A. (2021). Prediction of solar energy guided by pearson correlation using machine learning. *Energy*, *224*, 120109. https://doi.org/10.1016/j.energy.2021.120109

[21] Papaioannou, N., Myllis, G., Tsimpiris, A., & Vrana, V. (2025). The Role of Mutual Information Estimator Choice in Feature Selection: An Empirical Study on mRMR. *Information*, *16*(9), 724. https://doi.org/10.3390/info16090724

[22] Ali, M., Rabehi, A., Souahlia, A., Guermoui, M., Teta, A., Tibermacine, I. E., Agajie, T. F. (2025). Enhancing PV power forecasting through feature selection and artificial neural networks: A case study. https://doi.org/10.1038/s41598-025-07038-x

[23] Mucci, T. (2025, 17 noviembre). Data leakage machine learning. *What is data leakage in machine learning?* https://www.ibm.com/think/topics/data-leakage-machine-learning

[24] Alturayeif, N., & Hassine, J. (2025). Data leakage detection in machine learning code: transfer learning, active learning, or low-shot prompting?. *PeerJ. Computer science*, *11*, e2730. https://doi.org/10.7717/peerj-cs.2730

[25] Hasan, S. M. (2025, 18 enero). *Avoiding Data Leakage in Timeseries 101*. Towards Data Science. https://towardsdatascience.com/avoiding-data-leakage-in-timeseries-101-25ea13fcb15f/

[26] *11. Common pitfalls and recommended practices*. (s. f.). Scikit-learn. https://scikit-learn.org/stable/common_pitfalls.html

[27] Pelletier, H. (2025, 21 enero). *Cyclical Encoding: An Alternative to One-Hot Encoding for Time Series Features*. Towards Data Science. https://towardsdatascience.com/cyclical-encoding-an-alternative-to-one-hot-encoding-for-time-series-features-4db46248ebba/

[28] Javier. (2024, 9 octubre). *La importancia del perfilado de datos en Machine Learning*. Datadope. https://datadope.io/la-importancia-del-perfilado-de-datos-en-machine-learning/

[29] DataCamp. (2025). Preprocesamiento de datos: Una guía completa con ejemplos en Python. DataCamp. https://www.datacamp.com/es/blog/data-preprocessing

[30] Buhl, N. (2026, 4 febrero). *Training, Validation, Test Split for Machine Learning Datasets*. https://encord.com/blog/train-val-test-split/

[31] GeeksforGeeks. (2025, 24 octubre). *StandardScaler, MinMaxScaler and RobustScaler techniques ML*. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/

[32] Lin, Y., Koprinska, I., & Rana, M. (2021). Temporal convolutional neural networks for solar power forecasting. *IEEE Xplore. https://yanglin1997.github.io/files/TCNN.pdf*

[33] GeeksforGeeks. (2025). *What is Adam Optimizer?* GeeksforGeeks. https://www.geeksforgeeks.org/deep-learning/adam-optimizer/

[34] Amazon Web Services. (s. f.). *¿En qué consiste el ajuste de hiperparámetros?* Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/hyperparameter-tuning/

ANNEXES

Annex A. Methods for selecting exogenous variables

The following are some intermediate and advanced methods in the literature for the selection of exogenous variables.

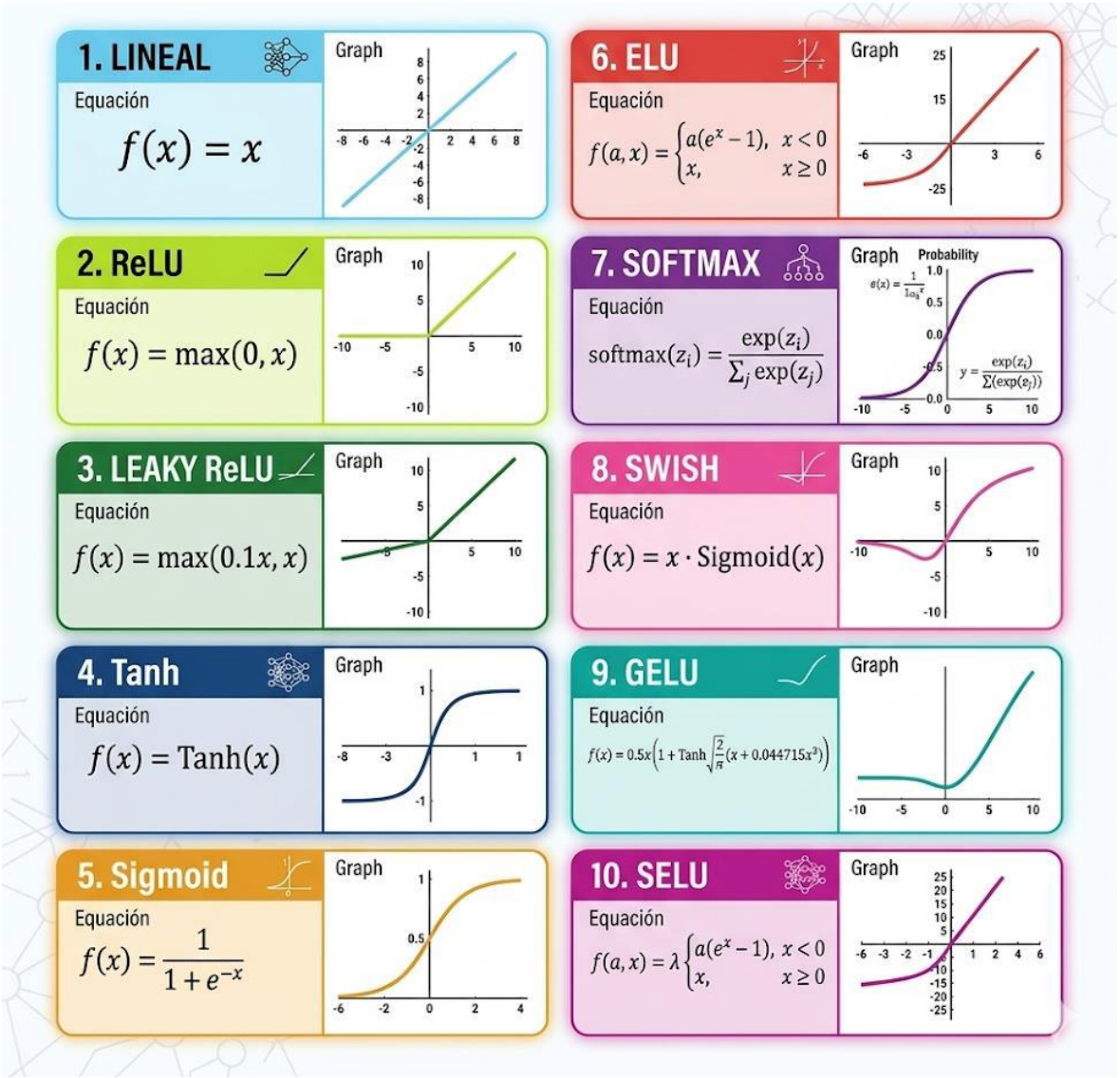
- Mutual information: It is an entropy-based measure that detects linear and nonlinear dependencies. It measures the amount of information that two random variables share and, specifically, how much uncertainty about a variable is reduced when the other variable is known [21].
- Relief technique: It is a robust technique utilized in machine learning and data mining to identify relevant features for classification tasks. It systematically evaluates the discriminatory power of each feature by analyzing the differences in feature values between instances of the same and different classes [22].
- Minimum correlation technique: The minimum correlation features selection (minimum CFS) technique is a straightforward and efficient method used for feature selection in Machine Learning. It operates by analyzing the correlations between input features to identify the most relevant predictors for a given task [22].
- Chi-square test technique: It is a statistical method used for assessing the association between categorical variables. It is particularly useful for analyzing contingency tables with two or more categories. The technique evaluates the independence between two variables by comparing observed frequencies to expected frequencies under the assumption of independence [22]. This technique is applicable when the candidate predictors or target-related groupings are categorical; otherwise, its use is limited in this context.
- F-test technique: F-test is a statistical procedure that computes an f-score by comparing variances. A higher f-score suggests that the differences within groups are minimal compared to the differences between groups [22].
- Neighborhood component analysis technique (NCA): It is a feature selection technique that aims to optimize the prediction accuracy of regression algorithms by learning a distance metric. NCA transforms input data to maximize average leave-one-out (LOO) classification performance. It addresses the drawbacks of the K-nearest neighbors (KNN) algorithm by utilizing a quadratic distance metric, reducing dimensionality, and improving computational efficiency [22]. NCA can also be

adapted to supervised feature weighting, but its use should be framed according to whether the task is classification or regression.

Annex B. Activation Functions

The following shows a representation of the activation functions both mathematically and visually.

Figure 9. Activation functions represented mathematically and visually



Note. Source: Generated using Google Gemini.

Annex C. Hyperparameters and configurations for the created models

This appendix addresses the details of the models, such as the optimal hyperparameters found for each model, as well as some important configurations used during model training that were not mentioned. It should be noted that, based on the tables created in Excel, an additional ablation study or significant conclusions regarding the selection of optimal hyperparameters could be presented; Although these tables contain additional details about the hyperparameters, this appendix is limited to reporting on the selected configurations.

Figure 10. Model features, epochs and settings for individual LSTM models.

		Summary for LSTM Models with Robust Scaler								
		Bayesian optimization/Random search settings			Epochs		Model features			
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons LSTM	Neurons Dense	Dropouts	Learning rate
POA1	2	100	20	729	40	70	64, 128	128	0,3 - 0,3	0,0005
	3	100	20	729	40	70	64, 32	96	0,1 - 0,2	0,001
	4	100	20	729	40	70	64, 32	96	0,2 - 0,3	0,005
	5	100	20	729	40	70	256, 32	128	0,3 - 0,1	0,001
POA2	2	100	20	729	40	70	64, 64	128	0,2 - 0,2	0,001
	3	100	20	729	40	70	128, 64	96	0,2 - 0,3	0,001
	4	100	20	729	40	70	64, 32	64	0,1 - 0,1	0,005
	5	100	20	729	40	70	64, 32	64	0,3 - 0,2	0,005
Rear1	2	100	20	729	40	70	128, 128	128	0,2 - 0,2	0,001
	3	100	20	729	40	70	128, 128	128	0,2 - 0,1	0,001
	4	100	20	729	40	70	256, 64	96	0,1 - 0,3	0,0005
	5	100	20	729	40	70	256, 128	64	0,3 - 0,3	0,0005
Rear2	2	100	20	729	40	70	128, 32	128	0,3 - 0,3	0,001
	3	100	20	729	40	70	64, 32	64	0,1 - 0,1	0,005
	4	100	20	729	40	70	64, 128	96	0,2 - 0,1	0,001
	5	100	20	729	40	70	64, 64	64	0,2 - 0,3	0,005
Rear3	2	100	20	729	40	70	64, 64	96	0,2 - 0,2	0,001
	3	100	20	729	40	70	64, 128	96	0,3 - 0,2	0,001
	4	100	20	729	40	70	64, 32	96	0,1 - 0,2	0,0005
	5	100	20	729	40	70	128, 128	128	0,1 - 0,3	0,001

		Summary for LSTM Models with MixMaxScaler								
		Bayesian optimization/Random search settings			Epochs		Model Features			
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons LSTM	Neurons Dense	Dropouts	Learning rate
POA1	2	100	20	729	40	70	64, 32	128	0,2 - 0,1	0,005
	3	100	20	729	40	70	128, 32	64	0,1 - 0,1	0,001
	4	100	20	729	40	70	64, 32	96	0,2 - 0,1	0,001
	5	100	20	729	40	70	128, 32	96	0,1 - 0,1	0,001
POA2	2	100	20	729	40	70	128, 128	64	0,3 - 0,2	0,001
	3	100	20	729	40	70	64, 32	128	0,2 - 0,1	0,001
	4	100	20	729	40	70	64, 32	64	0,1 - 0,1	0,001
	5	100	20	729	40	70	128, 32	128	0,1 - 0,1	0,001
Rear1	2	100	20	729	40	70	128, 128	96	0,1 - 0,2	0,0005
	3	100	20	729	40	70	64, 32	96	0,2 - 0,1	0,0005
	4	100	20	729	40	70	256, 32	128	0,3 - 0,2	0,005
	5	100	20	729	40	70	128, 64	64	0,1 - 0,3	0,005
Rear2	2	100	20	729	40	70	64, 32	64	0,2 - 0,2	0,005
	3	100	20	729	40	70	256, 64	96	0,3 - 0,1	0,001
	4	100	20	729	40	70	256, 64	96	0,3 - 0,2	0,001
	5	100	20	729	40	70	256, 32	128	0,2 - 0,1	0,001
Rear3	2	100	20	729	40	70	128, 32	96	0,1 - 0,1	0,001
	3	100	20	729	40	70	128, 32	64	0,2 - 0,3	0,001
	4	100	20	729	40	70	64, 32	96	0,1 - 0,2	0,0005
	5	100	20	729	40	70	256, 32	96	0,3 - 0,2	0,0005

Figure 11. Model features, epochs and settings for integrated LSTM Models.

		Summary for integrated LSTM Models with Robust Scaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons LSTM	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	150	30	1296	50	100	64, 96	128	192	0,3 - 0,3	0,0010
	3	150	30	1296	50	100	64, 64	128	192	0,2 - 0,2	0,001
	4	150	30	1296	50	100	320, 64	64	192	0,2 - 0,3	0,001
	5	150	30	1296	50	100	320, 128	96	192	0,3 - 0,2	0,0005
POA2	2	150	30	1296	50	100	64, 96	128	192	0,3 - 0,3	0,0010
	3	150	30	1296	50	100	64, 64	128	192	0,2 - 0,2	0,001
	4	150	30	1296	50	100	320, 64	64	192	0,2 - 0,3	0,001
	5	150	30	1296	50	100	320, 128	96	192	0,3 - 0,2	0,0005
Rear1	2	150	30	1296	50	100	64, 96	128	192	0,3 - 0,3	0,0010
	3	150	30	1296	50	100	64, 64	128	192	0,2 - 0,2	0,001
	4	150	30	1296	50	100	320, 64	64	192	0,2 - 0,3	0,001
	5	150	30	1296	50	100	320, 128	96	192	0,3 - 0,2	0,0005
Rear2	2	150	30	1296	50	100	64, 96	128	192	0,3 - 0,3	0,0010
	3	150	30	1296	50	100	64, 64	128	192	0,2 - 0,2	0,001
	4	150	30	1296	50	100	320, 64	64	192	0,2 - 0,3	0,001
	5	150	30	1296	50	100	320, 128	96	192	0,3 - 0,2	0,0005
Rear3	2	150	30	1296	50	100	64, 96	128	192	0,3 - 0,3	0,0010
	3	150	30	1296	50	100	64, 64	128	192	0,2 - 0,2	0,001
	4	150	30	1296	50	100	320, 64	64	192	0,2 - 0,3	0,001
	5	150	30	1296	50	100	320, 128	96	192	0,3 - 0,2	0,0005
		Summary for integrated LSTM Models with MixMaxScaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons LSTM	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	150	30	1296	50	100	320, 96	128	128	0,2 - 0,3	0,001
	3	150	30	1296	50	100	128, 128	96	96	0,2 - 0,3	0,0005
	4	150	30	1296	50	100	320, 128	96	96	0,2 - 0,2	0,0005
	5	150	30	1296	50	100	256, 64	64	96	0,3 - 0,2	0,0005
POA2	2	150	30	1296	50	100	320, 96	128	128	0,2 - 0,3	0,001
	3	150	30	1296	50	100	128, 128	96	96	0,2 - 0,3	0,0005
	4	150	30	1296	50	100	320, 128	96	96	0,2 - 0,2	0,0005
	5	150	30	1296	50	100	256, 64	64	96	0,3 - 0,2	0,0005
Rear1	2	150	30	1296	50	100	320, 96	128	128	0,2 - 0,3	0,001
	3	150	30	1296	50	100	128, 128	96	96	0,2 - 0,3	0,0005
	4	150	30	1296	50	100	320, 128	96	96	0,2 - 0,2	0,0005
	5	150	30	1296	50	100	256, 64	64	96	0,3 - 0,2	0,0005
Rear2	2	150	30	1296	50	100	320, 96	128	128	0,2 - 0,3	0,001
	3	150	30	1296	50	100	128, 128	96	96	0,2 - 0,3	0,0005
	4	150	30	1296	50	100	320, 128	96	96	0,2 - 0,2	0,0005
	5	150	30	1296	50	100	256, 64	64	96	0,3 - 0,2	0,0005
Rear3	2	150	30	1296	50	100	320, 96	128	128	0,2 - 0,3	0,001
	3	150	30	1296	50	100	128, 128	96	96	0,2 - 0,3	0,0005
	4	150	30	1296	50	100	320, 128	96	96	0,2 - 0,2	0,0005
	5	150	30	1296	50	100	256, 64	64	96	0,3 - 0,2	0,0005

Figure 12. Model features, epochs and settings for individual GRU models.

		Summary for GRU Models with Robust Scaler								
		Bayesian optimization/Random search settings			Epochs		Model Features			
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons GRU	Neurons Dense	Dropouts	Learning rate
POA1	2	90	18	486	40	70	64, 32	64	0,2 - 0,3	0,0050
	3	90	18	486	40	70	128, 32	96	0,1 - 0,1	0,001
	4	90	18	486	40	70	64, 32	64	0,1 - 0,3	0,005
	5	90	18	486	40	70	128, 128	64	0,3 - 0,3	0,001
POA2	2	90	18	486	40	70	64, 32	64	0,1 - 0,2	0,005
	3	90	18	486	40	70	128, 64	96	0,3 - 0,1	0,001
	4	90	18	486	40	70	64, 32	64	0,2 - 0,3	0,005
	5	90	18	486	40	70	64, 128	128	0,3 - 0,1	0,005
Rear1	2	90	18	486	40	70	64, 128	128	0,2 - 0,3	0,001
	3	90	18	486	40	70	128, 128	128	0,3 - 0,2	0,001
	4	90	18	486	40	70	64, 128	96	0,1 - 0,2	0,0005
	5	90	18	486	40	70	64, 128	128	0,1 - 0,2	0,005
Rear2	2	90	18	486	40	70	64, 32	128	0,1 - 0,3	0,005
	3	90	18	486	40	70	64, 32	64	0,3 - 0,2	0,005
	4	90	18	486	40	70	64, 128	128	0,1 - 0,3	0,005
	5	90	18	486	40	70	64, 32	96	0,1 - 0,3	0,001
Rear3	2	90	18	486	40	70	64, 64	64	0,1 - 0,3	0,005
	3	90	18	486	40	70	64, 32	96	0,3 - 0,1	0,001
	4	90	18	486	40	70	128, 128	64	0,3 - 0,3	0,001
	5	90	18	486	40	70	128, 128	96	0,3 - 0,1	0,001

		Summary for GRU Models with MixMaxScaler								
		Bayesian optimization/Random search settings			Epochs		Model Features			
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons GRU	Neurons Dense	Dropouts	Learning rate
POA1	2	90	18	486	40	70	128, 32	128	0,3 - 0,2	0,005
	3	90	18	486	40	70	128, 32	96	0,1 - 0,1	0,001
	4	90	18	486	40	70	128, 128	96	0,1 - 0,3	0,005
	5	90	18	486	40	70	128, 128	128	0,3 - 0,2	0,005
POA2	2	90	18	486	40	70	128, 128	96	0,1 - 0,3	0,005
	3	90	18	486	40	70	64, 32	96	0,3 - 0,1	0,005
	4	90	18	486	40	70	128, 128	96	0,1 - 0,1	0,005
	5	90	18	486	40	70	128, 128	64	0,1 - 0,3	0,005
Rear1	2	90	18	486	40	70	128, 32	64	0,3 - 0,2	0,005
	3	90	18	486	40	70	128, 32	64	0,3 - 0,2	0,001
	4	90	18	486	40	70	64, 64	64	0,3 - 0,3	0,001
	5	90	18	486	40	70	64, 32	96	0,2 - 0,1	0,001
Rear2	2	90	18	486	40	70	128, 32	96	0,1 - 0,1	0,005
	3	90	18	486	40	70	64, 128	96	0,2 - 0,3	0,001
	4	90	18	486	40	70	64, 128	128	0,3 - 0,3	0,001
	5	90	18	486	40	70	64,128	96	0,3 - 0,2	0,001
Rear3	2	90	18	486	40	70	128, 64	96	0,2 - 0,1	0,005
	3	90	18	486	40	70	64, 128	128	0,2 - 0,3	0,001
	4	90	18	486	40	70	128, 64	128	0,3 - 0,2	0,001
	5	90	18	486	40	70	64, 64	128	0,2 - 0,3	0,001

Figure 13. Model features, epochs and settings for integrated GRU Models.

		Summary for integrated GRU Models with Robust Scaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons GRU	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	28	140	972	50	100	256, 64	96	192	0,2 - 0,2	0,0010
	3	28	140	972	50	100	256, 64	96	128	0,3 - 0,3	0,0005
	4	28	140	972	50	100	256, 96	96	128	0,3 - 0,3	0,001
	5	28	140	972	50	100	256, 128	96	96	0,3 - 0,2	0,001
POA2	2	28	140	972	50	100	256, 64	96	192	0,2 - 0,2	0,0010
	3	28	140	972	50	100	256, 64	96	128	0,3 - 0,3	0,0005
	4	28	140	972	50	100	256, 96	96	128	0,3 - 0,3	0,001
	5	28	140	972	50	100	256, 128	96	96	0,3 - 0,2	0,001
Rear1	2	28	140	972	50	100	256, 64	96	192	0,2 - 0,2	0,0010
	3	28	140	972	50	100	256, 64	96	128	0,3 - 0,3	0,0005
	4	28	140	972	50	100	256, 96	96	128	0,3 - 0,3	0,001
	5	28	140	972	50	100	256, 128	96	96	0,3 - 0,2	0,001
Rear2	2	28	140	972	50	100	256, 64	96	192	0,2 - 0,2	0,0010
	3	28	140	972	50	100	256, 64	96	128	0,3 - 0,3	0,0005
	4	28	140	972	50	100	256, 96	96	128	0,3 - 0,3	0,001
	5	28	140	972	50	100	256, 128	96	96	0,3 - 0,2	0,001
Rear3	2	28	140	972	50	100	256, 64	96	192	0,2 - 0,2	0,0010
	3	28	140	972	50	100	256, 64	96	128	0,3 - 0,3	0,0005
	4	28	140	972	50	100	256, 96	96	128	0,3 - 0,3	0,001
	5	28	140	972	50	100	256, 128	96	96	0,3 - 0,2	0,001

		Summary for integrated GRU Models with MixMaxScaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons GRU	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	28	140	972	50	100	256, 96	128	128	0,2 - 0,3	0,001
	3	28	140	972	50	100	64, 64	96	96	0,2 - 0,3	0,001
	4	28	140	972	50	100	128, 128	96	96	0,2 - 0,2	0,001
	5	28	140	972	50	100	128, 64	128	96	0,2 - 0,2	0,001
POA2	2	28	140	972	50	100	256, 96	128	128	0,2 - 0,3	0,001
	3	28	140	972	50	100	64, 64	96	96	0,2 - 0,3	0,001
	4	28	140	972	50	100	128, 128	96	96	0,2 - 0,2	0,001
	5	28	140	972	50	100	128, 64	128	96	0,2 - 0,2	0,001
Rear1	2	28	140	972	50	100	256, 96	128	128	0,2 - 0,3	0,001
	3	28	140	972	50	100	64, 64	96	96	0,2 - 0,3	0,001
	4	28	140	972	50	100	128, 128	96	96	0,2 - 0,2	0,001
	5	28	140	972	50	100	128, 64	128	96	0,2 - 0,2	0,001
Rear2	2	28	140	972	50	100	256, 96	128	128	0,2 - 0,3	0,001
	3	28	140	972	50	100	64, 64	96	96	0,2 - 0,3	0,001
	4	28	140	972	50	100	128, 128	96	96	0,2 - 0,2	0,001
	5	28	140	972	50	100	128, 64	128	96	0,2 - 0,2	0,001
Rear3	2	28	140	972	50	100	256, 96	128	128	0,2 - 0,3	0,001
	3	28	140	972	50	100	64, 64	96	96	0,2 - 0,3	0,001
	4	28	140	972	50	100	128, 128	96	96	0,2 - 0,2	0,001
	5	28	140	972	50	100	128, 64	128	96	0,2 - 0,2	0,001

Figure 14. Model features, epochs and settings for individual BiLSTM models.

		Summary for BiLSTM Models with Robust Scaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiLSTM	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	20	100	729	40	70	64, 32	96	-	0,2 - 0,2	0,0010
	3	20	100	729	40	70	32, 16	96	-	0,2 - 0,2	0,0005
	4	20	100	729	40	70	32, 16	128	-	0,2 - 0,2	0,001
	5	20	100	729	40	70	128, 16	96	-	0,2 - 0,4	0,001
POA2	2	20	100	729	40	70	32, 64	128	-	0,3 - 0,4	0,001
	3	20	100	729	40	70	32, 16	128	-	0,4 - 0,2	0,001
	4	20	100	729	40	70	32, 32	96	-	0,2 - 0,3	0,001
	5	20	100	729	40	70	32, 16	96	-	0,4 - 0,3	0,001
Rear1	2	22	110	1458	40	70	128, 32	64	128	0,2 - 0,2	0,001
	3	22	110	1458	40	70	128, 64	64	96	0,4 - 0,2	0,0005
	4	22	110	1458	40	70	128, 64	96	96	0,2 - 0,3	0,001
	5	22	110	1458	40	70	128, 16	64	96	0,2 - 0,3	0,001
Rear2	2	22	110	1458	40	70	128, 32	128	96	0,3 - 0,4	0,001
	3	22	110	1458	40	70	128, 64	128	128	0,2 - 0,4	0,001
	4	22	110	1458	40	70	32, 16	128	128	0,3 - 0,2	0,001
	5	22	110	1458	40	70	32, 16	128	96	0,4 - 0,4	0,001
Rear3	2	22	110	1458	40	70	64, 64	96	128	0,2 - 0,4	0,001
	3	22	110	1458	40	70	128, 32	64	128	0,2 - 0,4	0,001
	4	22	110	1458	40	70	32, 64	96	128	0,3 - 0,4	0,001
	5	22	110	1458	40	70	32, 64	128	128	0,4 - 0,3	0,0005

		Summary for BiLSTM Models with MixMaxScaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiLSTM	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	20	100	729	40	70	128, 16	96	-	0,2 - 0,2	0,0005
	3	20	100	729	40	70	128, 16	96	-	0,3 - 0,2	0,001
	4	20	100	729	40	70	128, 16	96	-	0,4 - 0,2	0,001
	5	20	100	729	40	70	128, 16	96	-	0,2 - 0,2	0,001
POA2	2	20	100	729	40	70	128, 16	96	-	0,2 - 0,2	0,0005
	3	20	100	729	40	70	128, 16	128	-	0,2 - 0,2	0,001
	4	20	100	729	40	70	128, 16	128	-	0,2 - 0,2	0,001
	5	20	100	729	40	70	128, 16	96	-	0,2 - 0,2	0,001
Rear1	2	22	110	1458	40	70	128, 64	128	128	0,2 - 0,4	0,001
	3	22	110	1458	40	70	128, 32	128	96	0,4 - 0,4	0,0005
	4	22	110	1458	40	70	32, 16	128	128	0,4 - 0,3	0,001
	5	22	110	1458	40	70	32, 16	128	96	0,4 - 0,3	0,001
Rear2	2	22	110	1458	40	70	128, 16	64	96	0,3 - 0,2	0,001
	3	22	110	1458	40	70	64, 16	128	96	0,3 - 0,2	0,001
	4	22	110	1458	40	70	128, 64	128	96	0,2 - 0,2	0,001
	5	22	110	1458	40	70	64, 32	128	96	0,2 - 0,2	0,001
Rear3	2	22	110	1458	40	70	128, 16	128	96	0,3 - 0,4	0,001
	3	22	110	1458	40	70	64,16	64	128	0,4 - 0,4	0,001
	4	22	110	1458	40	70	128, 64	64	96	0,4 - 0,4	0,001
	5	22	110	1458	40	70	128, 64	128	96	0,4 - 0,4	0,001

Figure 15. Model features, epochs and settings for integrated BiLSTM Models.

		Summary for integrated BiLSTM Models with Robust Scaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiLSTM	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	28	140	864	50	100	128, 32	64	96	0,3 - 0,2	0,0010
	3	28	140	864	50	100	64, 64	96	128	0,2 - 0,3	0,001
	4	28	140	864	50	100	32, 32	96	128	0,2 - 0,2	0,001
	5	28	140	864	50	100	128, 32	96	192	0,3 - 0,3	0,001
POA2	2	28	140	864	50	100	128, 32	64	96	0,3 - 0,2	0,0010
	3	28	140	864	50	100	64, 64	96	128	0,2 - 0,3	0,001
	4	28	140	864	50	100	32, 32	96	128	0,2 - 0,2	0,001
	5	28	140	864	50	100	128, 32	96	192	0,3 - 0,3	0,001
Rear1	2	28	140	864	50	100	128, 32	64	96	0,3 - 0,2	0,0010
	3	28	140	864	50	100	64, 64	96	128	0,2 - 0,3	0,001
	4	28	140	864	50	100	32, 32	96	128	0,2 - 0,2	0,001
	5	28	140	864	50	100	128, 32	96	192	0,3 - 0,3	0,001
Rear2	2	28	140	864	50	100	128, 32	64	96	0,3 - 0,2	0,0010
	3	28	140	864	50	100	64, 64	96	128	0,2 - 0,3	0,001
	4	28	140	864	50	100	32, 32	96	128	0,2 - 0,2	0,001
	5	28	140	864	50	100	128, 32	96	192	0,3 - 0,3	0,001
Rear3	2	28	140	864	50	100	128, 32	64	96	0,3 - 0,2	0,0010
	3	28	140	864	50	100	64, 64	96	128	0,2 - 0,3	0,001
	4	28	140	864	50	100	32, 32	96	128	0,2 - 0,2	0,001
	5	28	140	864	50	100	128, 32	96	192	0,3 - 0,3	0,001

		Summary for integrated BiLSTM Models with MixMaxScaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiLSTM	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	28	140	864	50	100	64, 64	128	128	0,3 - 0,2	0,001
	3	28	140	864	50	100	64, 64	96	96	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 32	128	96	0,3 - 0,3	0,0005
	5	28	140	864	50	100	128, 32	128	128	0,3 - 0,2	0,0005
POA2	2	28	140	864	50	100	64, 64	128	128	0,3 - 0,2	0,001
	3	28	140	864	50	100	64, 64	96	96	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 32	128	96	0,3 - 0,3	0,0005
	5	28	140	864	50	100	128, 32	128	128	0,3 - 0,2	0,0005
Rear1	2	28	140	864	50	100	64, 64	128	128	0,3 - 0,2	0,001
	3	28	140	864	50	100	64, 64	96	96	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 32	128	96	0,3 - 0,3	0,0005
	5	28	140	864	50	100	128, 32	128	128	0,3 - 0,2	0,0005
Rear2	2	28	140	864	50	100	64, 64	128	128	0,3 - 0,2	0,001
	3	28	140	864	50	100	64, 64	96	96	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 32	128	96	0,3 - 0,3	0,0005
	5	28	140	864	50	100	128, 32	128	128	0,3 - 0,2	0,0005
Rear3	2	28	140	864	50	100	64, 64	128	128	0,3 - 0,2	0,001
	3	28	140	864	50	100	64, 64	96	96	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 32	128	96	0,3 - 0,3	0,0005
	5	28	140	864	50	100	128, 32	128	128	0,3 - 0,2	0,0005

Figure 16. Model features, epochs and settings for individual BiGRU models.

		Summary for BiGRU Models with Robust Scaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiGRU	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	16	80	216	40	70	64, 16	96	-	0,2 - 0,2	0,0005
	3	16	80	216	40	70	64, 16	96	-	0,2 - 0,2	0,001
	4	16	80	216	40	70	32, 32	64	-	0,2 - 0,3	0,001
	5	16	80	216	40	70	64, 32	64	-	0,2 - 0,2	0,001
POA2	2	16	80	216	40	70	32, 64	64	-	0,2 - 0,2	0,001
	3	16	80	216	40	70	64, 64	96	-	0,2 - 0,2	0,001
	4	16	80	216	40	70	32, 64	64	-	0,3 - 0,2	0,001
	5	16	80	216	40	70	32, 16	128	-	0,2 - 0,2	0,001
Rear1	2	18	90	432	40	70	64, 64	128	96	0,2 - 0,3	0,001
	3	18	90	432	40	70	64, 32	128	96	0,2 - 0,3	0,001
	4	18	90	432	40	70	64, 16	64	96	0,2 - 0,3	0,0005
	5	18	90	432	40	70	32, 16	128	96	0,2 - 0,3	0,0005
Rear2	2	18	90	432	40	70	32, 16	128	96	0,3 - 0,3	0,001
	3	18	90	432	40	70	64, 16	128	128	0,3 - 0,2	0,001
	4	18	90	432	40	70	64, 16	128	128	0,3 - 0,3	0,001
	5	18	90	432	40	70	64, 64	64	128	0,3 - 0,3	0,001
Rear3	2	18	90	432	40	70	64, 32	128	128	0,3 - 0,2	0,001
	3	18	90	432	40	70	64, 16	64	128	0,3 - 0,3	0,001
	4	18	90	432	40	70	64, 64	128	128	0,2 - 0,3	0,001
	5	18	90	432	40	70	64, 16	64	96	0,2 - 0,2	0,001

		Summary for BiGRU Models with MixMaxScaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiGRU	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	16	80	216	40	70	64, 16	64	-	0,3 - 0,3	0,001
	3	16	80	216	40	70	64, 32	128	-	0,2 - 0,2	0,001
	4	16	80	216	40	70	64, 64	128	-	0,3 - 0,2	0,001
	5	16	80	216	40	70	64, 32	96	-	0,2 - 0,2	0,001
POA2	2	16	80	216	40	70	32, 16	128	-	0,3 - 0,3	0,0005
	3	16	80	216	40	70	64, 64	96	-	0,2 - 0,2	0,001
	4	16	80	216	40	70	64, 64	96	-	0,2 - 0,2	0,001
	5	16	80	216	40	70	64, 64	64	-	0,2 - 0,2	0,001
Rear1	2	18	90	432	40	70	64, 32	128	128	0,2 - 0,2	0,001
	3	18	90	432	40	70	32, 32	128	96	0,2 - 0,3	0,0005
	4	18	90	432	40	70	64, 16	64	96	0,3 - 0,3	0,001
	5	18	90	432	40	70	32, 16	64	96	0,3 - 0,2	0,0005
Rear2	2	18	90	432	40	70	32, 16	128	96	0,3 - 0,3	0,0005
	3	18	90	432	40	70	32, 16	128	96	0,2 - 0,3	0,0005
	4	18	90	432	40	70	64, 16	128	128	0,2 - 0,3	0,001
	5	18	90	432	40	70	32, 16	128	96	0,3 - 0,2	0,001
Rear3	2	18	90	432	40	70	64, 64	128	96	0,2 - 0,2	0,001
	3	18	90	432	40	70	32, 64	128	96	0,2 - 0,2	0,001
	4	18	90	432	40	70	32, 32	128	96	0,2 - 0,2	0,001
	5	18	90	432	40	70	64, 64	128	96	0,2 - 0,2	0,001

.

Figure 17. Model features, epochs and settings for integrated BiGRU Models.

		Summary for integrated BiGRU Models with Robust Scaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiGRU	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	28	140	864	50	100	64, 32	128	128	0,2 - 0,3	0,0010
	3	28	140	864	50	100	128, 32	96	128	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 64	128	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	128, 64	128	96	0,3 - 0,3	0,001
POA2	2	28	140	864	50	100	64, 32	128	128	0,2 - 0,3	0,0010
	3	28	140	864	50	100	128, 32	96	128	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 64	128	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	128, 64	128	96	0,3 - 0,3	0,001
Rear1	2	28	140	864	50	100	64, 32	128	128	0,2 - 0,3	0,0010
	3	28	140	864	50	100	128, 32	96	128	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 64	128	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	128, 64	128	96	0,3 - 0,3	0,001
Rear2	2	28	140	864	50	100	64, 32	128	128	0,2 - 0,3	0,0010
	3	28	140	864	50	100	128, 32	96	128	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 64	128	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	128, 64	128	96	0,3 - 0,3	0,001
Rear3	2	28	140	864	50	100	64, 32	128	128	0,2 - 0,3	0,0010
	3	28	140	864	50	100	128, 32	96	128	0,3 - 0,3	0,001
	4	28	140	864	50	100	128, 64	128	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	128, 64	128	96	0,3 - 0,3	0,001
		Summary for integrated BiGRU Models with MixMaxScaler									
		Bayesian optimization/Random search settings			Epochs		Model Features				
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	Neurons BiGRU	Neurons Dense	Neurons concatenate	Dropouts	Learning rate
POA1	2	28	140	864	50	100	64, 64	128	128	0,2 - 0,3	0,001
	3	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
	4	28	140	864	50	100	32, 64	96	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
POA2	2	28	140	864	50	100	64, 64	128	128	0,2 - 0,3	0,001
	3	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
	4	28	140	864	50	100	32, 64	96	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
Rear1	2	28	140	864	50	100	64, 64	128	128	0,2 - 0,3	0,001
	3	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
	4	28	140	864	50	100	32, 64	96	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
Rear2	2	28	140	864	50	100	64, 64	128	128	0,2 - 0,3	0,001
	3	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
	4	28	140	864	50	100	32, 64	96	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
Rear3	2	28	140	864	50	100	64, 64	128	128	0,2 - 0,3	0,001
	3	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001
	4	28	140	864	50	100	32, 64	96	96	0,3 - 0,2	0,001
	5	28	140	864	50	100	32, 32	128	96	0,2 - 0,2	0,001

Figure 18. Model features, epochs and settings for individual TCN models.

		Summary for TCN Models with Robust Scaler										
		Bayesian optimization/Random search settings			Epochs		Model Features					
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropout	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	-	72	72	40	70	3	64	0,3	96	-	0,0010
	3	-	72	72	40	70	5	32	0,3	128	-	0,001
	4	-	72	72	40	70	3	64	0,2	128	-	0,001
	5	-	72	72	40	70	3	64	0,2	128	-	0,001
POA2	2	-	72	72	40	70	5	32	0,3	128	-	0,001
	3	-	72	72	40	70	3	64	0,2	64	-	0,001
	4	-	72	72	40	70	3	64	0,2	128	-	0,001
	5	-	72	72	40	70	5	32	0,2	128	-	0,001
Rear1	2	16	80	144	40	70	5	64	0,3	96	128	0,001
	3	16	80	144	40	70	3	32	0,2	96	128	0,001
	4	16	80	144	40	70	3	64	0,2	64	96	0,001
	5	16	80	144	40	70	3	64	0,2	64	128	0,0005
Rear2	2	16	80	144	40	70	3	32	0,3	96	128	0,001
	3	16	80	144	40	70	3	64	0,3	64	128	0,001
	4	16	80	144	40	70	5	64	0,2	128	128	0,001
	5	16	80	144	40	70	5	32	0,2	128	128	0,001
Rear3	2	16	80	144	40	70	5	32	0,2	128	128	0,001
	3	16	80	144	40	70	3	32	0,3	96	128	0,001
	4	16	80	144	40	70	3	64	0,3	64	96	0,001
	5	16	80	144	40	70	5	64	0,2	64	128	0,001

		Summary for TCN Models with MixMaxScaler										
		Bayesian optimization/Random search settings			Epochs		Model Features					
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropout	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	-	72	72	40	70	5	64	0,2	128	-	0,001
	3	-	72	72	40	70	3	64	0,2	128	-	0,001
	4	-	72	72	40	70	5	32	0,2	128	-	0,001
	5	-	72	72	40	70	3	64	0,2	96	-	0,001
POA2	2	-	72	72	40	70	5	64	0,2	128	-	0,001
	3	-	72	72	40	70	5	64	0,2	96	-	0,001
	4	-	72	72	40	70	3	64	0,2	96	-	0,001
	5	-	72	72	40	70	5	64	0,3	128	-	0,001
Rear1	2	16	80	144	40	70	5	64	0,3	64	96	0,001
	3	16	80	144	40	70	5	64	0,3	64	128	0,001
	4	16	80	144	40	70	5	64	0,2	64	96	0,001
	5	16	80	144	40	70	3	64	0,2	64	96	0,001
Rear2	2	16	80	144	40	70	3	64	0,2	64	96	0,001
	3	16	80	144	40	70	3	32	0,3	128	96	0,001
	4	16	80	144	40	70	5	64	0,2	64	96	0,001
	5	16	80	144	40	70	3	32	0,3	64	96	0,001
Rear3	2	16	80	144	40	70	5	32	0,2	128	128	0,001
	3	16	80	144	40	70	3	32	0,2	128	128	0,001
	4	16	80	144	40	70	3	32	0,3	96	128	0,001
	5	16	80	144	40	70	3	32	0,2	128	96	0,001

Figure 19. Model features, epochs and settings for integrated TCN Models.

		Summary for integrated TCN Models with Robust Scaler										
		Bayesian optimization/Random search settings			Epochs		Model Features					
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropout	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	20	100	324	50	100	5	64	0,3	96	96	0,0010
	3	20	100	324	50	100	5	32	0,3	128	192	0,001
	4	20	100	324	50	100	3	32	0,2	128	192	0,001
	5	20	100	324	50	100	3	32	0,2	128	192	0,001
POA2	2	20	100	324	50	100	5	64	0,3	96	96	0,0010
	3	20	100	324	50	100	5	32	0,3	128	192	0,001
	4	20	100	324	50	100	3	32	0,2	128	192	0,001
	5	20	100	324	50	100	3	32	0,2	128	192	0,001
Rear1	2	20	100	324	50	100	5	64	0,3	96	96	0,0010
	3	20	100	324	50	100	5	32	0,3	128	192	0,001
	4	20	100	324	50	100	3	32	0,2	128	192	0,001
	5	20	100	324	50	100	3	32	0,2	128	192	0,001
Rear2	2	20	100	324	50	100	5	64	0,3	96	96	0,0010
	3	20	100	324	50	100	5	32	0,3	128	192	0,001
	4	20	100	324	50	100	3	32	0,2	128	192	0,001
	5	20	100	324	50	100	3	32	0,2	128	192	0,001
Rear3	2	20	100	324	50	100	5	64	0,3	96	96	0,0010
	3	20	100	324	50	100	5	32	0,3	128	192	0,001
	4	20	100	324	50	100	3	32	0,2	128	192	0,001
	5	20	100	324	50	100	3	32	0,2	128	192	0,001

		Summary for integrated TCN Models with MixMaxScaler										
		Bayesian optimization/Random search settings			Epochs		Model Features					
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropout	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	20	100	324	50	100	5	64	0,2	96	128	0,001
	3	20	100	324	50	100	5	64	0,3	64	192	0,001
	4	20	100	324	50	100	3	64	0,2	64	96	0,001
	5	20	100	324	50	100	3	64	0,2	64	192	0,001
POA2	2	20	100	324	50	100	5	64	0,2	96	128	0,001
	3	20	100	324	50	100	5	64	0,3	64	192	0,001
	4	20	100	324	50	100	3	64	0,2	64	96	0,001
	5	20	100	324	50	100	3	64	0,2	64	192	0,001
Rear1	2	20	100	324	50	100	5	64	0,2	96	128	0,001
	3	20	100	324	50	100	5	64	0,3	64	192	0,001
	4	20	100	324	50	100	3	64	0,2	64	96	0,001
	5	20	100	324	50	100	3	64	0,2	64	192	0,001
Rear2	2	20	100	324	50	100	5	64	0,2	96	128	0,001
	3	20	100	324	50	100	5	64	0,3	64	192	0,001
	4	20	100	324	50	100	3	64	0,2	64	96	0,001
	5	20	100	324	50	100	3	64	0,2	64	192	0,001
Rear3	2	20	100	324	50	100	5	64	0,2	96	128	0,001
	3	20	100	324	50	100	5	64	0,3	64	192	0,001
	4	20	100	324	50	100	3	64	0,2	64	96	0,001
	5	20	100	324	50	100	3	64	0,2	64	192	0,001

Figure 20. Model features, epochs and settings for individual CNN-LSTM models.

		Summary for CNN-LSTM Models with Robust Scaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons LSTM	Neurons Dense	Neurons Concatenate	Learning rate
POA1	2	16	80	192	40	70	5	32	0,1 - 0,1	128	96	-	0,0005
	3	16	80	192	40	70	3	32	0,1 - 0,1	128	96	-	0,0005
	4	16	80	192	40	70	5	64	0,1 - 0,1	128	96	-	0,001
	5	16	80	192	40	70	5	64	0,2 - 0,1	128	96	-	0,0005
POA2	2	16	80	192	40	70	3	32	0,2 - 0,1	96	96	-	0,0005
	3	16	80	192	40	70	3	64	0,1 - 0,2	64	96	-	0,0005
	4	16	80	192	40	70	3	64	0,1 - 0,2	64	64	-	0,001
	5	16	80	192	40	70	5	32	0,1 - 0,1	128	96	-	0,0005
Rear1	2	18	90	384	40	70	3	32	0,2 - 0,2	128	96	128	0,0005
	3	18	90	384	40	70	3	64	0,1 - 0,2	128	96	96	0,0005
	4	18	90	384	40	70	3	64	0,2 - 0,2	96	64	96	0,0005
	5	18	90	384	40	70	5	32	0,1 - 0,2	128	64	128	0,0005
Rear2	2	18	90	384	40	70	5	32	0,2 - 0,2	128	96	128	0,001
	3	18	90	384	40	70	3	32	0,2 - 0,1	128	64	128	0,0005
	4	18	90	384	40	70	5	32	0,1 - 0,2	128	64	96	0,0005
	5	18	90	384	40	70	5	64	0,1 - 0,1	128	96	128	0,0005
Rear3	2	18	90	384	40	70	5	32	0,1 - 0,2	128	64	96	0,001
	3	18	90	384	40	70	5	64	0,2 - 0,1	64	96	96	0,001
	4	18	90	384	40	70	3	32	0,2 - 0,1	128	64	128	0,0005
	5	18	90	384	40	70	3	32	0,2 - 0,1	128	64	128	0,0005

		Summary for CNN-LSTM Models with MixMaxScaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons LSTM	Neurons Dense	Neurons Concatenate	Learning rate
POA1	2	16	80	192	40	70	5	64	0,2 - 0,1	128	96	-	0,0005
	3	16	80	192	40	70	3	64	0,1 - 0,1	64	64	-	0,001
	4	16	80	192	40	70	3	32	0,1 - 0,2	96	64	-	0,001
	5	16	80	192	40	70	3	32	0,1 - 0,2	128	96	-	0,0005
POA2	2	16	80	192	40	70	3	64	0,1 - 0,1	128	96	-	0,001
	3	16	80	192	40	70	3	64	0,1 - 0,2	64	96	-	0,0005
	4	16	80	192	40	70	3	64	0,2 - 0,2	64	64	-	0,001
	5	16	80	192	40	70	5	32	0,1 - 0,1	128	96	-	0,0005
Rear1	2	18	90	384	40	70	3	64	0,1 - 0,2	64	64	128	0,001
	3	18	90	384	40	70	3	64	0,2 - 0,2	96	96	96	0,0005
	4	18	90	384	40	70	5	32	0,2 - 0,2	64	64	128	0,0005
	5	18	90	384	40	70	3	32	0,2 - 0,2	64	96	96	0,0005
Rear2	2	18	90	384	40	70	3	32	0,1 - 0,1	64	64	96	0,001
	3	18	90	384	40	70	3	32	0,1 - 0,2	64	64	96	0,001
	4	18	90	384	40	70	5	64	0,2 - 0,2	128	96	96	0,001
	5	18	90	384	40	70	3	32	0,1 - 0,1	128	96	128	0,0005
Rear3	2	18	90	384	40	70	5	64	0,1 - 0,1	128	96	96	0,001
	3	18	90	384	40	70	3	32	0,1 - 0,1	96	96	96	0,0005
	4	18	90	384	40	70	3	32	0,2 - 0,1	128	64	128	0,0005
	5	18	90	384	40	70	5	32	0,1 - 0,2	128	96	128	0,001

Figure 21. Model features, epochs and settings for integrated CNN-LSTM Models.

		Summary for integrated CNN-LSTM Models with Robust Scaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons LSTM	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	28	140	864	50	100	5	64	0,2 - 0,2	96	128	128	0,0010
	3	28	140	864	50	100	3	64	0,1 - 0,2	96	64	96	0,0005
	4	28	140	864	50	100	3	64	0,2 - 0,2	128	64	128	0,0005
	5	28	140	864	50	100	5	32	0,1 - 0,1	96	128	96	0,0005
POA2	2	28	140	864	50	100	5	64	0,2 - 0,2	96	128	128	0,0010
	3	28	140	864	50	100	3	64	0,1 - 0,2	96	64	96	0,0005
	4	28	140	864	50	100	3	64	0,2 - 0,2	128	64	128	0,0005
	5	28	140	864	50	100	5	32	0,1 - 0,1	96	128	96	0,0005
Rear1	2	28	140	864	50	100	5	64	0,2 - 0,2	96	128	128	0,0010
	3	28	140	864	50	100	3	64	0,1 - 0,2	96	64	96	0,0005
	4	28	140	864	50	100	3	64	0,2 - 0,2	128	64	128	0,0005
	5	28	140	864	50	100	5	32	0,1 - 0,1	96	128	96	0,0005
Rear2	2	28	140	864	50	100	5	64	0,2 - 0,2	96	128	128	0,0010
	3	28	140	864	50	100	3	64	0,1 - 0,2	96	64	96	0,0005
	4	28	140	864	50	100	3	64	0,2 - 0,2	128	64	128	0,0005
	5	28	140	864	50	100	5	32	0,1 - 0,1	96	128	96	0,0005
Rear3	2	28	140	864	50	100	5	64	0,2 - 0,2	96	128	128	0,0010
	3	28	140	864	50	100	3	64	0,1 - 0,2	96	64	96	0,0005
	4	28	140	864	50	100	3	64	0,2 - 0,2	128	64	128	0,0005
	5	28	140	864	50	100	5	32	0,1 - 0,1	96	128	96	0,0005

		Summary for integrated CNN-LSTM Models with MixMaxScaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons LSTM	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	28	140	864	50	100	3	32	0,1 - 0,2	64	64	192	0,001
	3	28	140	864	50	100	3	64	0,2 - 0,2	96	96	96	0,001
	4	28	140	864	50	100	5	32	0,1 - 0,1	64	128	96	0,0005
	5	28	140	864	50	100	3	64	0,2 - 0,1	128	64	96	0,001
POA2	2	28	140	864	50	100	3	32	0,1 - 0,2	64	64	192	0,001
	3	28	140	864	50	100	3	64	0,2 - 0,2	96	96	96	0,001
	4	28	140	864	50	100	5	32	0,1 - 0,1	64	128	96	0,0005
	5	28	140	864	50	100	3	64	0,2 - 0,1	128	64	96	0,001
Rear1	2	28	140	864	50	100	3	32	0,1 - 0,2	64	64	192	0,001
	3	28	140	864	50	100	3	64	0,2 - 0,2	96	96	96	0,001
	4	28	140	864	50	100	5	32	0,1 - 0,1	64	128	96	0,0005
	5	28	140	864	50	100	3	64	0,2 - 0,1	128	64	96	0,001
Rear2	2	28	140	864	50	100	3	32	0,1 - 0,2	64	64	192	0,001
	3	28	140	864	50	100	3	64	0,2 - 0,2	96	96	96	0,001
	4	28	140	864	50	100	5	32	0,1 - 0,1	64	128	96	0,0005
	5	28	140	864	50	100	3	64	0,2 - 0,1	128	64	96	0,001
Rear3	2	28	140	864	50	100	3	32	0,1 - 0,2	64	64	192	0,001
	3	28	140	864	50	100	3	64	0,2 - 0,2	96	96	96	0,001
	4	28	140	864	50	100	5	32	0,1 - 0,1	64	128	96	0,0005
	5	28	140	864	50	100	3	64	0,2 - 0,1	128	64	96	0,001

Figure 22. Model features, epochs and settings for individual CNN-GRU models.

		Summary for CNN-GRU Models with Robust Scaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons GRU	Neurons Dense	Neurons Concatenate	Learning rate
POA1	2	16	80	192	40	70	5	32	0,2 - 0,2	32	64	-	0,0010
	3	16	80	192	40	70	5	32	0,2 - 0,2	96	64	-	0,001
	4	16	80	192	40	70	5	64	0,2 - 0,1	96	96	-	0,0005
	5	16	80	192	40	70	5	64	0,2 - 0,1	96	96	-	0,0005
POA2	2	16	80	192	40	70	3	32	0,1 - 0,1	96	64	-	0,001
	3	16	80	192	40	70	5	32	0,2 - 0,1	96	96	-	0,0005
	4	16	80	192	40	70	3	64	0,2 - 0,1	32	96	-	0,0005
	5	16	80	192	40	70	5	64	0,2 - 0,1	64	96	-	0,001
Rear1	2	18	90	384	40	70	3	64	0,2 - 0,1	96	96	96	0,001
	3	18	90	384	40	70	3	64	0,2 - 0,1	96	96	96	0,001
	4	18	90	384	40	70	3	64	0,2 - 0,2	96	64	128	0,001
	5	18	90	384	40	70	3	32	0,2 - 0,2	96	64	128	0,001
Rear2	2	18	90	384	40	70	5	32	0,1 - 0,2	32	96	96	0,0005
	3	18	90	384	40	70	3	32	0,1 - 0,1	64	96	128	0,0005
	4	18	90	384	40	70	5	32	0,1 - 0,2	32	96	96	0,0005
	5	18	90	384	40	70	3	32	0,2 - 0,2	32	96	96	0,0005
Rear3	2	18	90	384	40	70	5	32	0,2 - 0,2	32	96	128	0,0005
	3	18	90	384	40	70	5	32	0,2 - 0,1	32	64	128	0,0005
	4	18	90	384	40	70	3	32	0,1 - 0,2	32	96	96	0,0005
	5	18	90	384	40	70	3	32	0,1 - 0,1	32	64	96	0,0005

		Summary for CNN-GRU Models with MixMaxScaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons GRU	Neurons Dense	Neurons Concatenate	Learning rate
POA1	2	16	80	192	40	70	5	32	0,2 - 0,2	96	96	-	0,001
	3	16	80	192	40	70	5	64	0,1 - 0,2	32	64	-	0,001
	4	16	80	192	40	70	3	32	0,1 - 0,2	64	64	-	0,001
	5	16	80	192	40	70	3	64	0,2 - 0,2	96	96	-	0,0005
POA2	2	16	80	192	40	70	3	32	0,2 - 0,1	64	64	-	0,001
	3	16	80	192	40	70	3	32	0,2 - 0,2	32	96	-	0,001
	4	16	80	192	40	70	3	32	0,1 - 0,2	64	96	-	0,0005
	5	16	80	192	40	70	3	64	0,1 - 0,2	32	96	-	0,0005
Rear1	2	18	90	384	40	70	3	64	0,1 - 0,2	32	96	96	0,001
	3	18	90	384	40	70	5	64	0,2 - 0,2	32	96	96	0,001
	4	18	90	384	40	70	3	32	0,1 - 0,2	32	96	96	0,001
	5	18	90	384	40	70	3	32	0,1 - 0,2	96	96	128	0,001
Rear2	2	18	90	384	40	70	5	32	0,1 - 0,1	32	96	96	0,001
	3	18	90	384	40	70	3	64	0,1 - 0,2	96	64	128	0,0005
	4	18	90	384	40	70	5	32	0,1 - 0,1	64	64	96	0,001
	5	18	90	384	40	70	5	32	0,1 - 0,1	96	96	128	0,001
Rear3	2	18	90	384	40	70	3	64	0,2 - 0,2	96	64	96	0,0005
	3	18	90	384	40	70	5	64	0,1 - 0,2	32	96	96	0,0005
	4	18	90	384	40	70	3	32	0,1 - 0,1	64	64	96	0,001
	5	18	90	384	40	70	5	32	0,1 - 0,1	96	96	128	0,001

Figure 23. Model features, epochs and settings for integrated CNN-GRU Models.

		Summary for integrated CNN-GRU Models with Robust Scaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons GRU	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	28	140	864	50	100	5	64	0,2 - 0,1	64	64	96	0,0010
	3	28	140	864	50	100	5	32	0,2 - 0,1	64	96	192	0,001
	4	28	140	864	50	100	3	32	0,1 - 0,1	96	128	128	0,0005
	5	28	140	864	50	100	5	64	0,1 - 0,2	96	128	192	0,0005
POA2	2	28	140	864	50	100	5	64	0,2 - 0,1	64	64	96	0,0010
	3	28	140	864	50	100	5	32	0,2 - 0,1	64	96	192	0,001
	4	28	140	864	50	100	3	32	0,1 - 0,1	96	128	128	0,0005
	5	28	140	864	50	100	5	64	0,1 - 0,2	96	128	192	0,0005
Rear1	2	28	140	864	50	100	5	64	0,2 - 0,1	64	64	96	0,0010
	3	28	140	864	50	100	5	32	0,2 - 0,1	64	96	192	0,001
	4	28	140	864	50	100	3	32	0,1 - 0,1	96	128	128	0,0005
	5	28	140	864	50	100	5	64	0,1 - 0,2	96	128	192	0,0005
Rear2	2	28	140	864	50	100	5	64	0,2 - 0,1	64	64	96	0,0010
	3	28	140	864	50	100	5	32	0,2 - 0,1	64	96	192	0,001
	4	28	140	864	50	100	3	32	0,1 - 0,1	96	128	128	0,0005
	5	28	140	864	50	100	5	64	0,1 - 0,2	96	128	192	0,0005
Rear3	2	28	140	864	50	100	5	64	0,2 - 0,1	64	64	96	0,0010
	3	28	140	864	50	100	5	32	0,2 - 0,1	64	96	192	0,001
	4	28	140	864	50	100	3	32	0,1 - 0,1	96	128	128	0,0005
	5	28	140	864	50	100	5	64	0,1 - 0,2	96	128	192	0,0005

		Summary for integrated CNN-GRU Models with MixMaxScaler											
		Bayesian optimization/Random search settings			Epochs		Model Features						
Sensor	n	Initial points	Max_trials	Total combinations	epochs	epochs for retraining	K_Size	filters	Dropouts	Neurons GRU	Neurons Dense	Neurons concatenate	Learning rate
POA1	2	28	140	864	50	100	5	32	0,1 - 0,2	96	96	96	0,0005
	3	28	140	864	50	100	5	64	0,2 - 0,2	32	64	96	0,0005
	4	28	140	864	50	100	3	32	0,1 - 0,1	64	96	96	0,001
	5	28	140	864	50	100	3	32	0,1 - 0,2	64	96	128	0,0005
POA2	2	28	140	864	50	100	5	32	0,1 - 0,2	96	96	96	0,0005
	3	28	140	864	50	100	5	64	0,2 - 0,2	32	64	96	0,0005
	4	28	140	864	50	100	3	32	0,1 - 0,1	64	96	96	0,001
	5	28	140	864	50	100	3	32	0,1 - 0,2	64	96	128	0,0005
Rear1	2	28	140	864	50	100	5	32	0,1 - 0,2	96	96	96	0,0005
	3	28	140	864	50	100	5	64	0,2 - 0,2	32	64	96	0,0005
	4	28	140	864	50	100	3	32	0,1 - 0,1	64	96	96	0,001
	5	28	140	864	50	100	3	32	0,1 - 0,2	64	96	128	0,0005
Rear2	2	28	140	864	50	100	5	32	0,1 - 0,2	96	96	96	0,0005
	3	28	140	864	50	100	5	64	0,2 - 0,2	32	64	96	0,0005
	4	28	140	864	50	100	3	32	0,1 - 0,1	64	96	96	0,001
	5	28	140	864	50	100	3	32	0,1 - 0,2	64	96	128	0,0005
Rear3	2	28	140	864	50	100	5	32	0,1 - 0,2	96	96	96	0,0005
	3	28	140	864	50	100	5	64	0,2 - 0,2	32	64	96	0,0005
	4	28	140	864	50	100	3	32	0,1 - 0,1	64	96	96	0,001
	5	28	140	864	50	100	3	32	0,1 - 0,2	64	96	128	0,0005

Annex D. Complete Ablation Analysis

This annex presents the complete study of ablation, its explainability, comparisons and analysis of results for this project, which includes the analysis of performance by quartiles and sensors, the effect of the scalers, the differences between individual and integrated models, as well as the use or not of retraining and the impact on exogenous variables.

For the recurrent architectures (Group 1) and Q25, these results were obtained:

- POA1 and Rear 1: BiGRU and GRU showed the best average performance. LSTM and BiLSTM stood out in specific cases.
- POA2: GRU and BiLSTM achieved the best average MAE, while BiGRU and BiLSTM led in R^2.
- Rear 2: LSTM and GRU led in MAE, while LSTM and BiLSTM stood out in R^2.
- Rear 3: BiGRU and GRU maintained the best overall average performance.

Likewise, for Q50 (easy and intermediate days) the results obtained are:

- POA1 and Rear 3: BiGRU and GRU consistently achieved the best average performance in MAE, R^2, and MAPE.
- POA2: BiGRU and BiLSTM delivered the best average results across all three metrics.
- Rear 1: BiLSTM and GRU led in MAE; BiGRU and BiLSTM in R^2; and GRU and BiGRU together in MAPE.
- Rear 2: BiLSTM and BiGRU outperformed in R^2 and MAPE, while GRU outperformed BiGRU in MAE.

Similarly, for Q75 (easy, intermediate and some difficult days) the results obtained are:

- POA1: BiGRU and GRU led the average, although LSTM outperformed BiLSTM in MAE and R^2 for single models without retraining.
- POA2: GRU outperformed BiLSTM in R^2 and MAPE metrics.
- Rear 1: BiGRU and GRU performed best in R^2 and MAPE, but BiLSTM outperformed BiGRU in MAE.

- Rear 2: LSTM and GRU led in MAE, while BiGRU and GRU were the best in R^2 and MAPE.
- Rear 3: BiGRU and GRU continued to demonstrate the best overall average performance.

Finally, for Q100 (i.e., the complete test set), GRU and BiGRU showed the strongest MAE performance across POA1, POA2, and Rear sensors. Regarding the scaler used for our group 1, it was obtained.

- MinMax Scaler: This scaler was optimal for the vast majority of cases in the Q25 and Q50 quartiles. Its use is critical for maximizing performance on the POA1 and POA2 sensors.
- Robust Scaler: Its relevance increases proportionally with the difficulty of the quartile. For Q75, models using Robust Scaler outperformed MinMax on the Rear 1 sensor and showed better overall MAE on the Rear 2 sensor.

For integrated and individual models, the impact of retraining is concluded as follows:

- No Retraining: This configuration consistently produced better R^2 and MAPE in most scenarios for the POA and Rear sensors.
- With Retraining: Retraining was only advantageous in specific cases of individual models with MinMax on the Rear 2 sensor and, generally, for improving performance on the Rear 1 sensor.
- Individual Models: Demonstrated superior R^2 on the front sensors (POA1 and POA2).
- Integrated Models: Were more effective on the rear sensors, especially when combined with Robust Scaler for the Q50 quartile.

To conclude the analysis of Group 1, the effect of exogenous variables is summarized below.

- Front Sensors (POA1 and POA2):
    - General Trend (Q25 and Q50): A substantial improvement in performance is observed when increasing the number of exogenous variables, especially when using MinMax Scaler.

    - Architecture Specifics: The BiLSTM architecture shows linear improvement up to 5 variables in individual models. But the GRU reaches its maximum efficiency point with 3 variables, after which its performance stabilizes
    - Technical Restriction: In these sensors, the use of Robust Scaler has an adverse effect; adding external variables degrades the accuracy of all architectures in the group.

- Rear Sensors (Rear 1 and Rear 2):
    - Behavior in Q50 and Q75: The Rear 1 sensor benefits the most from the added complexity, especially in models that incorporate retraining.
    - Optimization in Rear 2: For Q75 difficulty scenarios, the LSTM architecture (in single configuration with Robust Scaler) better captures the R^2 variance as the amount of exogenous information increases.

- Rear 3 Sensor: This sensor exhibits early saturation behavior. No significant improvement patterns were identified when exceeding two exogenous variables.

Similarly, the ablation analysis for the second group is presented, starting with the analysis by quartiles and sensors; for Q25, the following is obtained:

- POA1 and POA2: CNN-GRU demonstrated the greatest stability and the best average performance.
- Rear 1 and Rear 3: CNN-LSTM exhibited the greatest stability and the best average performance in most cases.
- Rear 2: CNN-GRU was the most stable, although both CNN-GRU and CNN-LSTM shared the lead in performance according to the evaluated metric.

Likewise, for Q50 (easy and intermediate days) the results obtained are:

- POA1 and POA2: CNN-GRU maintained the best overall average performance in MAE, R^2, and MAPE.
- Rear 1: CNN-GRU led in all metrics on average.
- Rear 2: CNN-LSTM outperformed in MAE and R^2, while CNN-GRU achieved the best MAPE.

- Rear 3: CNN-LSTM led in MAE and MAPE, while CNN-GRU excelled in R^2.

Similarly, for Q75 (easy, intermediate and some difficult days) the results obtained are:

- POA1 and POA2: CNN-LSTM outperformed the other architectures in R^2
- Rear 1: CNN-GRU achieved the best results in R^2 and MAPE, although CNN-LSTM outperformed MAPE in certain scenarios.
- Rear 2: CNN-LSTM was the best architecture on average across all metrics.
- Rear 3: CNN-LSTM maintained its lead in MAE and MAPE, while CNN-GRU outperformed in R^2.

Finally, for Q100 (All the test days), All sensors (POA1, POA2, Rear 1, 2, and 3): As in Group 1, overall performance was low on critical days. However, MAE analysis for all days positioned CNN-GRU and CNN-LSTM as the architectures with the strongest performance across all front and rear sensors. Regarding scaling strategies, the main findings are as follows:

- Robust Scaler: Unlike Group 1, for Group 2 architectures, the Robust Scaler was superior in most cases for the Q50 and Q75 quartiles, especially in MAE.
- MinMax scaler: This scaling technique only showed superiority in specific scenarios.

For integrated and individual models, the impact of retraining is concluded as follows:

- No Retraining: The absence of dynamic retraining resulted in better R^2 and MAPE values in the vast majority of tests for Q50 and Q75.
- With Retraining: This practice was only beneficial in very specific cases.
- Individual Models: These models performed better in R^2 for the POA sensors, especially in the Q75 quartile.
- Integrated Models: These models demonstrated better performance on the rear sensors (Rear 2 and Rear 3) for the Q50 quartile.

Regarding the dynamics of exogenous variables for Group 2, the following conclusions were reached.

- Front sensors (POA1 and POA2):
    - In Q50 and Q75: These architectures demonstrate a superior capacity for processing external information. Performance (MAE and R^2) improves progressively up to 5 exogenous variables.
    - Exception in TCN: TCN shows limited benefits when combined with MinMax Scaler and retraining, suggesting that its temporal structure prefers static training data.

- Rear sensors (Rear 1 and Rear 2):
    - Dominance of Robust Scaler (Q50): The combination of CNN-LSTM or CNN-GRU with Robust Scaler allows for a drastic reduction in error when adding meteorological variables, finding an optimal balance between 3 and 5 variables.
    - Stability in Q75: The use of exogenous information helps mitigate the loss of accuracy on challenging days for these specific sensors.

- Rear sensor 3: Consistency across all quartiles: As in Group 1, this sensor does not respond significantly to increased variables. Models achieve their best metric with 2 exogenous variables

Annex E. Code Repository

The entire project was developed in sprints, and each sprint had a challenge and some related code. Therefore, all the code created was organized and divided by sprint in a code repository called Development-of-the-MALEBISOPA-project. All the code was written in Python, and environments such as Google Colab and Visual Studio were used throughout the project. Here is the link to the repository where the code is located.

https://github.com/jmanurodri04-glitch/Development-of-the-MALEBISOPA-project