energies

MDPI

Article

Energy Management of Industrial Energy Systems via Rolling Horizon and Hybrid Optimization: A Real-Plant Application in Germany

Loukas Kyriakidis *, Rushit Kansara and Maria Isabel Roldán Serrano *

German Aerospace Center, Institute of Low-Carbon Industrial Processes, Simulation and Virtual Design Department, Walther-Pauer-Straße 5, 03046 Cottbus, Germany; rushit.kansara@dlr.de
* Correspondence: loukas.kyriakidis@dlr.de (L.K.); maria.roldanserrano@dlr.de (M.I.R.S.)

Abstract

Industrial energy systems are increasingly required to reduce operating costs and CO_2 emissions while integrating variable renewable energy sources. Managing these objectives under uncertainty requires advanced optimization strategies capable of delivering reliable and real-time decisions. To address these challenges, this study focuses on the short-term operational planning of an industrial energy supply system using the rolling horizon approach (RHA). The RHA offers an effective framework to handle uncertainties by repeatedly updating forecasts and re-optimizing over a moving time window, thereby enabling adaptive and responsive energy management. To solve the resulting nonlinear and constrained optimization problem at each RHA iteration, we propose a novel hybrid algorithm that combines Bayesian optimization (BO) with the Interior Point OPTimizer (IPOPT). While global deterministic and stochastic optimization methods are frequently used in practice, they often suffer from high computational costs and slow convergence, particularly when applied to large-scale, nonlinear problems with complex constraints. To overcome these limitations, we employ the BO–IPOPT, integrating the global search capabilities of BO with the efficient local convergence and constraint fulfillment of the IPOPT. Applied to a large-scale real-world case study of a food and cosmetic industry in Germany, the proposed BO–IPOPT method outperformed state-of-the-art solvers in both solution quality and robustness, achieving up to 97.25%-better objective function values at the same CPU time. Additionally, the influence of key parameters, such as forecast uncertainty, optimization horizon length, and computational effort per RHA iteration, was analyzed to assess their impact on system performance and decision quality.

Keywords: food and cosmetic industry; energy management; rolling horizon approach; BO–IPOPT

check for updates

Academic Editor: Brian D. Fath

Received: 19 June 2025
Revised: 15 July 2025
Accepted: 23 July 2025
Published: 25 July 2025

Citation: Kyriakidis, L.; Kansara, R.; Roldán Serrano, M.I. Energy Management of Industrial Energy Systems via Rolling Horizon and Hybrid Optimization: A Real-Plant Application in Germany. *Energies* **2025**, *18*, 3977. https://doi.org/10.3390/en18153977

1. Introduction

Energy supply systems play a crucial role in modern infrastructure, and their efficient management is crucial in addressing today's global energy challenges. As the world moves toward cleaner and more sustainable energy sources, the structure and operation of these systems have become increasingly complex [1,2]. Key goals include reducing operating costs, improving overall efficiency, and reducing CO_2 emissions [2]. However, integrating renewable energy technologies introduces new challenges, such as dealing with fluctuating energy generation and changing demand patterns, while maintaining grid stability and power quality [3]. The variability of renewable sources also makes accurate

forecasting more difficult, which adds complexity to operational planning. As a result, advanced energy management strategies that can effectively handle uncertainties have become essential for ensuring reliable and sustainable system performance.

To address these uncertainties in the energy systems, various methodologies have been developed, each with its own strengths and applications. Stochastic programming, for instance, models uncertain parameters using known probability distributions, enabling probabilistic decision-making under uncertainty [4]. In contrast, robust optimization avoids reliance on probabilistic data and instead aims to find solutions that remain feasible and effective across a defined set of scenarios, including worst-case conditions [5]. Fuzzy logic provides another alternative, particularly useful when dealing with imprecise or vague information, by incorporating degrees of truth into decision processes [6]. Monte Carlo simulations offer yet another strategy by generating a wide range of possible outcomes through repeated sampling of random variables, thus allowing for statistical analysis of risk and uncertainty [7].

Although the aforementioned methods offer structured ways to handle uncertainties, they often come with high computational demands. This becomes problematic in systems that require adaptation to fast-changing conditions. To overcome this limitation, the RHA has gained widespread use, particularly for applications requiring real-time response [8]. The RHA operates by repeatedly solving an optimization problem over a finite horizon in the future. After implementing the first step of the resulting plan, the time window shifts forward, and the process is repeated with newly updated data. This iterative and forward-looking structure makes the RHA especially advantageous for managing renewable energy systems, where unpredictability in conditions requires regular updates of energy management decisions based on new forecasts and operational data. For this reason, the RHA was considered for addressing uncertainties in this work.

The effectiveness of the RHA in managing energy systems under uncertainty is closely linked to the underlying optimization methods. These techniques enable optimal operational decisions in response to updated forecasts and system conditions. Within the RHA framework, a wide range of optimization approaches has been applied to address both linear and nonlinear problem types. For linear formulations, mixed-integer linear programming (MILP) has mainly been employed, due to its ability to model both discrete and continuous variables efficiently [8–12], while linear programming (LP), limited to modeling only continuous variables, plays a minor role in the literature [13,14]. Although linear and linearized models benefit from reduced computational time and scalability, they often compromise model fidelity. Linearization, in particular, demands careful tuning to strike an effective trade-off between accuracy and computational speed.

In contrast, nonlinear optimization has seen more limited use in RHA-based energy management, particularly when it comes to deterministic global solvers such as BARON [15], which tend to be computationally expensive in high-dimensional, nonconvex problems. Nevertheless, some studies have implemented global stochastic approaches like genetic algorithms (GA) and particle swarm optimization (PSO) [16,17] to tackle nonlinear problems in microgrids and hybrid renewable systems. These techniques offer the advantage of escaping local optima, but they typically demand a large number of function evaluations, making them less practical for large-scale, constrained problems.

To address the computational and convergence limitations of conventional stochastic solvers, this work integrates the recently developed BO–IPOPT algorithm into the RHA for energy management of industrial energy systems. The BO–IPOPT is a hybrid optimization technique—originally presented in [18] and further enhanced in [19]—that leverages the strengths of BO for global search and the IPOPT for efficient local convergence and constraint fulfillment. Prior studies have demonstrated its superior performance in solv-

ing complex, high-dimensional, and nonlinear constrained optimization problems with improved accuracy and robustness over existing methods, achieving up to 120%-better objective function values at the same CPU time.

In this study, we applied the BO–IPOPT to a real-world industrial energy system at a food and cosmetics facility consisting of a solar thermal, photovoltaics, heat pump, and three stratified thermal storages. To support the system's operational planning, which depends on accurate solar forecasts, we also conducted a comparative analysis of several data-driven forecasting techniques at two German locations, employing a recursive multi-step-ahead approach. This comparison aimed to identify forecasting models that not only achieve high prediction accuracy but also maintain low computational overhead, which is essential for their integration into real-time optimization routines in practical applications.

Beyond evaluating the performance of the BO–IPOPT against other state-of-the-art optimization methods, this paper investigates how variations in optimizer CPU running time, forecast uncertainty, and planning horizons influence the operation of the investigated industrial energy system. Through this analysis, we provide practical insights into the real-world applicability of the BO–IPOPT framework within the RHA, contributing to the development of more sustainable and responsive energy management strategies aimed at minimizing operating costs and reducing CO_2 emissions in the underlying case study.

This paper is organized as follows: Section 2 describes the industrial use case on which this study is based. Section 3 presents the hybrid optimization method "BO–IPOPT" employed at each step of the RHA. Section 4 investigates the performance of several data-based models for solar irradiance forecasting across two locations in Germany, analyzing both one-step and multi-step prediction scenarios. Section 5 presents the outcomes of the energy management strategy for the modeled energy system. Finally, Section 6 concludes the paper with key findings and outlines directions for future research.

2. Use Case

In this section, we present the industrial energy system investigated in this work, including a detailed description of its architecture, the modeling approach used for each of its components, and the formulation of the resulting operational optimization problem.

2.1. Description of System

In this study, a company located in Herzberg in the German state of Brandenburg was analyzed as a use case. The company operates in the food and cosmetics industrial sector. In line with its goal to decarbonize its heat and electricity demand, the company had recently undergone a comprehensive retrofit of its energy supply infrastructure, incorporating additional components to enable the use of renewable energy sources. The energy management of the resulting system—displayed in Figure 1—is analyzed in this study, with the objective of minimizing both operating costs and CO_2 emissions through optimal system operation.

The energy system analyzed in this study represents a comprehensive integration of renewable energy sources and thermal energy storage (TES) technologies designed to decarbonize the facility's energy supply. Central to this configuration is a solar thermal collector (STC), which charges two stratified TES units. These TES units are hydraulically linked to the STC, whereby low-temperature water from the lower layers of the TESs is supplied to the STC inlet. After solar heating, the water is returned to the upper layers of the TESs, maintaining thermal stratification and TES efficiency.

The energy stored in these two tanks is used in multiple ways. A portion of the heat is directed to the heat source and sink of a heat pump (HP), which is powered by electricity supplied from an on-site photovoltaic (PV) system and, when necessary, the electrical grid.

The HP's source side also draws thermal energy from a separate hot water tank, which itself is partially charged using heat from the initial two TES units. If the PV generation exceeds the site's consumption, the excess electricity can be sold to the market, supporting the grid integration of renewables.

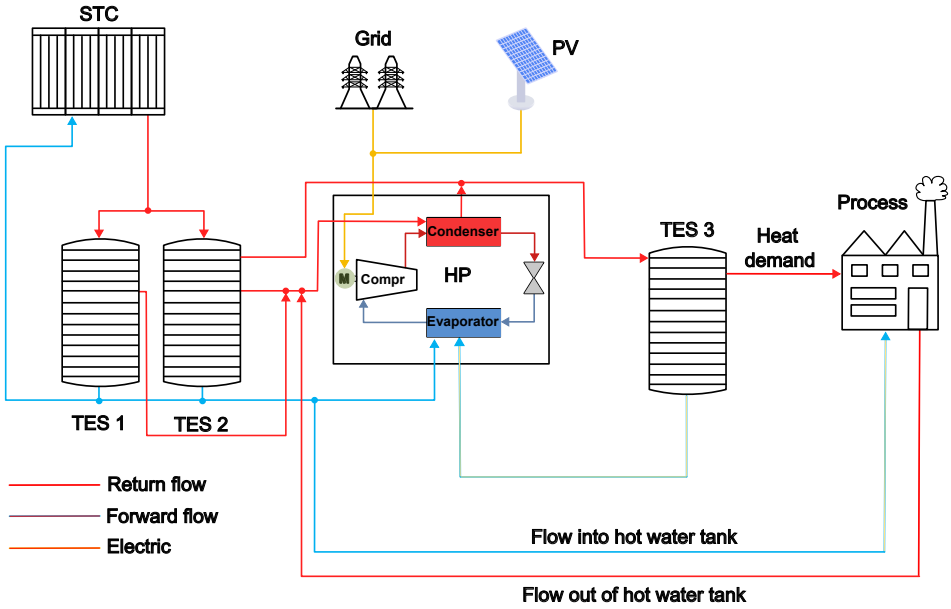

Figure 1. Visualization of the industrial energy system.

The HP upgrades low-temperature thermal energy to higher temperatures (up to 90 °C) suitable for industrial use. Its output is directed to a third stratified TES unit. This third TES serves as the main thermal buffer for the industrial process, supplying the required mass flow (3000–3500 L/h) and temperature (88–90 °C) of hot water. This corresponds to a maximum thermal power output of approximately 260 kW. Additionally, it is designed to receive excess heat from the first two TES units when the temperature in their upper layers is sufficiently high. In contrast, during periods of low solar input or high demand the lower layers of the third TES can supply heat back to the HP's source side, enhancing the operational flexibility. The stratified thermal storage units operate across a temperature range of approximately 25 °C to 90 °C, allowing for efficient thermal layering and flexible charging/discharging strategies.

Water is used throughout the entire process as the heat transfer and TES medium, due to its compatibility with the system components.

2.2. Component Modeling

To evaluate the performance and optimize the operation of the renewable energy system introduced in the previous subsection, detailed component modeling was necessary. This included mathematical representations of the STC, PV, HP, and TES units. These models formed the basis of the optimization problem and were designed to capture physical behavior.

2.2.1. Solar Thermal Collector

The STC was modeled to estimate the thermal energy output as a function of the incident solar irradiance, collector area, optical and thermal characteristics, and relevant temperature differences. The thermal output of the collector is defined as follows [20]:

$$\dot{m}_{\mathrm{STC}} c_{p,\mathrm{water}}(T_{\mathrm{STC,in}}, T_{\mathrm{STC,out}})(T_{\mathrm{STC,out}} - T_{\mathrm{STC,in}}) = (1 - \mathrm{bypass}_{\mathrm{STC}}) A_{\mathrm{STC}} \left[K_{\mathrm{STC}} I \eta_{\mathrm{STC,opt}} - \sum_{i=0}^{4} \beta_{\mathrm{STC},i} \Delta T^{i}_{\mathrm{STC,amb}} \right] \quad (1)$$

where $\dot{m}_{\mathrm{STC}}$ is the mass flow rate through the collector, $c_{p,\mathrm{water}}(T_{\mathrm{STC,in}}, T_{\mathrm{STC,out}})$ represents the temperature-dependent specific heat capacity of water, $T_{\mathrm{STC,out}}$ and $T_{\mathrm{STC,in}}$ denote the outlet and inlet temperatures of the collector, and A_{STC} is the collector area. Moreover, I is the solar irradiance incident on the collector and $\eta_{\mathrm{STC,opt}}$ describes the optical efficiency. Thermal losses are represented by the polynomial terms involving $\beta_{\mathrm{STC},i}$, which are loss coefficients for different orders of the temperature difference. The factor $\mathrm{bypass}_{\mathrm{STC}}$ accounts for periods when the operation of the collector is restricted or deactivated.

The term K_{STC} represents the incidence-correction factor, which accounts for the angle of incidence of solar irradiance on the collector surface, and it is calculated by the following formula:

$$K_{\mathrm{STC}} = 1 - \frac{1 - \alpha_{\mathrm{STC}}}{0.55} \left(\frac{1}{\cos(\theta_{\mathrm{STC}})} - 1 \right) \quad (2)$$

where θ_{STC} is the angle of incidence and α_{STC} an empirical coefficient. The temperature difference between the collector and ambient, $\Delta T_{\mathrm{STC,amb}}$, is calculated as

$$\Delta T_{\mathrm{STC,amb}} = \frac{T_{\mathrm{STC,out}} + T_{\mathrm{STC,in}}}{2} - T_{\mathrm{amb}} \quad (3)$$

where T_{amb} is the ambient temperature. All the parameters, summarized in Table 1, were derived from the collector's technical specifications and [21]. The installed solar thermal collector system has a peak thermal capacity of approximately 100 kW.

Table 1. STC parameters.

Parameter	Value	Unit
$\dot{Q}_{\mathrm{STC,peak}}$	100	kW
A_{STC}	135	m^2
$\eta_{\mathrm{STC,opt}}$	0.715	1
α_{STC}	0.97	1
$\beta_{\mathrm{STC},0}$	1.1	1
$\beta_{\mathrm{STC},1}$	3.31	$W/(m^2K)$
$\beta_{\mathrm{STC},2}$	0.011	$W/(m^2K^2)$
θ_{STC}	32	$^\circ$

2.2.2. Photovoltaic

The PV system was modeled to estimate the PV power output P_{PV} as a function of the available solar irradiance, module area, and the efficiencies of both the inverter and the PV module. The PV output is given by

$$P_{\mathrm{PV}} = (1 - \mathrm{bypass}_{\mathrm{PV}}) A_{\mathrm{PV}} I \eta_{\mathrm{PV,mod}} \eta_{\mathrm{PV,inv}} \quad (4)$$

with

$$\eta_{\mathrm{PV,mod}} = \left(\eta_{\mathrm{PV,nom}} + a_{\mathrm{PV},1} \ln\left(\frac{I}{I_{\mathrm{PV,nom}}} \right) \right) (1 + a_{\mathrm{PV},2}(T_{\mathrm{amb}} + a_{\mathrm{PV},3} I - T_{\mathrm{PV,nom}})) \quad (5)$$

Here, A_{PV} is the module area, $\eta_{\mathrm{PV,inv}}$ is the inverter efficiency, and $\eta_{\mathrm{PV,nom}}$ denotes the nominal module efficiency. The parameters $a_{\mathrm{PV},1}$, $a_{\mathrm{PV},2}$, and $a_{\mathrm{PV},3}$ are empirical coefficients

used to capture the part-load behavior of the module under different irradiance and temperature conditions. Moreover, $I_{PV,nom}$ denotes the nominal solar irradiance, and $T_{PV,nom}$ the nominal temperature. The term $bypass_{PV}$ accounts for the restriction or deactivation of the PV operation. The remaining symbols are defined in the previous section. The parameters, summarized in Table 2, were derived from the design specifications applied in the experimental setup and supported by data from [22–25]. The installed PV system had a nominal peak power capacity of 55 kWp.

Table 2. PV parameters.

Variable	Value	Unit
A_{PV}	500	m^2
$P_{PV,peak}$	55	kWp
$\eta_{PV,nom}$	0.212	1
$\eta_{PV,inv}$	0.95	1
$a_{PV,1}$	0.06408	1
$a_{PV,2}$	−0.0035	1/K
$a_{PV,3}$	0.0275	K m^2/W
$T_{PV,nom}$	42	°C
$I_{PV,nom}$	1000	W/m^2

2.2.3. Heat Pump

The HP model relied on surrogate models constructed for two critical performance indicators: the outlet temperature on the sink side $T_{h,out}$ and the electrical power consumption P_{el}. These surrogate models were based on technical specifications provided by the manufacturer and were created using second-order polynomial regression.

The regression models performed well across the full operating range, including part-load conditions, with coefficients of determination (R^2) of 99.9% for the outlet temperature and 99.6% for the electrical power consumption. Key input variables included the inlet temperatures on both the source and sink sides ($T_{c,in}$,$T_{h,in}$), as well as the mass flow rate on the sink side $\dot{m}_{h,in}$. These inputs significantly affected the thermal and electrical performance of the HP. The surrogate model for the outlet temperature on the sink side was expressed as

$$T_{h,out} = f(T_{c,in}, T_{h,in}, \dot{m}_{h,in}, \dot{m}_{h,in}^2, T_{c,in}T_{h,in}, T_{c,in}\dot{m}_{h,in}, T_{h,in}\dot{m}_{h,in}) \quad (6)$$

Similarly, the electrical power consumption was modeled as

$$P_{el} = f(T_{c,in}, T_{h,in}, \dot{m}_{h,in}, T_{c,in}^2, T_{h,in}^2, \dot{m}_{h,in}^2, T_{c,in}T_{h,in}, T_{c,in}\dot{m}_{h,in}, T_{h,in}\dot{m}_{h,in}) \quad (7)$$

These surrogate models were integrated into the overall optimization framework, ensuring both physical fidelity and computational efficiency.

2.2.4. Thermal Energy Storage

Each TES in this work was modeled using a detailed physics-based approach categorized as a multi-node one-dimensional (1-D) model [26]. In this formulation, the tank is discretized into N vertical layers (or nodes), each assumed to have a uniform temperature T_i, representing vertical thermal gradients while neglecting horizontal variations. The model employs volume and energy balance equations for each node, capturing both mass flows and heat transfer dynamics. The general forms of the governing equations are given by

$$\frac{dV}{dt} = \sum \dot{V}(t) = 0 \quad (8)$$

$$\frac{dE}{dt} = \sum h(t)\dot{m}(t) + \sum \dot{Q} \tag{9}$$

Equation (8) ensures volume conservation in each node, while Equation (9) represents the energy conservation including enthalpy and external heat transfer between adjacent nodes and ambient losses. The dynamic mass and energy balances for a generic node i are expressed as

$$\dot{V}_{\text{vert,out}} = \frac{\dot{m}_{\text{in}}}{\rho(T_{\text{in}})} + \dot{V}_{\text{vert,in}} - \frac{\dot{m}_{\text{out}}}{\rho(T_i)} \tag{10}$$

$$\begin{aligned} m_i c_p(T_i)\frac{dT_i}{dt} = & \sum \dot{m}_{\text{in}} c_p(T_{\text{in}})T_{\text{in}} - \sum \dot{m}_{\text{out}} c_p(T_{\text{out}})T_{\text{out}} \\ & - \dot{V}_{\text{vert,out}}\rho(T_{\text{vert,out}})c_p(T_{\text{vert,out}})T_{\text{vert,out}} \\ & + \dot{V}_{\text{vert,in}}\rho(T_{\text{vert,in}})c_p(T_{\text{vert,in}})\dot{V}_{\text{vert,in}} \\ & + k(T_{i-1}, T_i)(T_{i-1} - T_i) - k(T_i, T_{i+1})(T_i - T_{i+1}) \\ & - UA(T_i - T_{\text{amb}}) \end{aligned} \tag{11}$$

Here, m_i is the mass of the fluid in node i; $c_p(T)$, $\rho(T)$, and $k(T)$ denote the temperature-dependent specific heat capacity, density, and thermal conductivity, respectively; and UA accounts for ambient heat losses, where U is the overall heat transfer coefficient of the insulation and A is the heat transfer surface area. In this work, an overall heat transfer $U = 0.043$ kW/(m^2 K) was assumed, and the surface area was taken as $A = 12.53$ m^2, based on the design of the storages. Each TES was discretized into four layers (nodes) in this study, where the node index i started from the top of the storage and increased downward. The choice of four layers represented a balance between capturing sufficient thermal stratification and enabling flexible charging/discharging behavior, while keeping the dimensions of the resulting optimization problem (see Section 2.3) computationally manageable. This level of discretization ensured appropriate resolution for modeling thermal gradients without introducing unnecessary computational complexity. The terms involving k describe thermal exchange between neighboring nodes, referred to as "pseudo-conduction" terms, which approximate convective mixing effects. The sign and temperature assignments for vertical flows were handled as follows, depending on the flow direction:

$$T_{\text{vert,out}} = T_i \quad \text{if} \quad \dot{V}_{\text{vert,out}} > 0 \tag{12}$$

$$T_{\text{vert,in}} = T_{i-1} \quad \text{if} \quad \dot{V}_{\text{vert,in}} > 0 \tag{13}$$

$$T_{\text{vert,out}} = T_{i+1} \quad \text{if} \quad \dot{V}_{\text{vert,out}} < 0 \tag{14}$$

$$T_{\text{vert,in}} = T_i \quad \text{if} \quad \dot{V}_{\text{vert,in}} < 0 \tag{15}$$

This modeling approach allowed the TES behavior to be accurately predicted in terms of stratification, charging/discharging dynamics, and thermal losses.

2.3. Optimization Problem

In this study, we consider an optimization problem for the optimal operation of an energy system under both economic and environmental objectives. The problem is formulated as a discrete-time, multi-period optimization over a finite planning horizon. The continuous time interval $[t_0, t_n]$ is discretized into n equidistant steps with a time step $\Delta t = t_k - t_{k-1}$ for $k = 1, \ldots, n$. All control variables and system outputs are assumed to be piecewise constant over each interval and are evaluated at discrete time points t_k.

The objective is formulated as a weighted sum of the operating costs and the associated CO_2 emissions. The operating costs result from electricity expenses due to grid power imports and are partially offset by revenues from excess PV electricity fed into the grid. CO_2

emissions are proportional to the grid electricity consumption, based on a fixed emission factor that reflects the carbon intensity of certified eco-electricity, including emissions associated with its provision and infrastructure. The complete objective function is given as

$$\begin{aligned} \min\ f(P_{\text{grid}}, P_{\text{PV,sell}}) = & w_{\text{cost}} \sum_{k=1}^{n} (P_{\text{grid}}^{k} g_{\text{pr,grid}} - P_{\text{PV,sell}}^{k} g_{\text{PV,sell}}) \Delta t + \\ & w_{\text{em}} \sum_{k=1}^{n} P_{\text{grid}}^{k} g_{\text{em,grid}} \Delta t \end{aligned} \tag{16}$$

Here, P_{grid}^{k} denotes the imported grid power at time step k, $P_{\text{PV,sell}}^{k}$ the power sold from PV to the grid, and Δt the time step duration. The electricity price $g_{\text{pr,grid}}$ is fixed at 0.17 €/kWh, the feed-in tariff $g_{\text{PV,sell}}$ at 0.16 €/kWh, and the emission factor $g_{\text{em,grid}}$ is set to 55.7 g/kWh. The scalar weights w_{cost} and w_{em} are both set to 0.5, reflecting the industry's equal emphasis on cost-efficiency and environmental responsibility. The optimization is subject to the following set of constraints:

- **Component models:** At each k, the behavior of the STC, PV, HP, and TES units must satisfy their respective mathematical models defined in Sections 2.2.1–2.2.4. It should be mentioned that the component models account for temperature-dependent material properties of water, which are essential for accurately modeling thermal processes [27]. The specific heat capacity is represented as a third-degree polynomial function of temperature, while the thermal conductivity and density are modeled using second-degree polynomial expressions.
- **Flow and connectivity constraints:** The mass and energy flows between the components are modeled at each k, using equality constraints to ensure that the conservation principles are satisfied at all connection points. The interconnections among the system components, including bypasses, have been described in Section 2.1. The general form of the mass and energy conservation equations at each connection point can be expressed as

$$\sum \dot{m}_{\text{in}}^{k} = \sum \dot{m}_{\text{out}}^{k} \quad \text{(mass conservation)} \tag{17}$$

$$\sum \dot{m}_{\text{in}}^{k} h_{\text{in}}^{k} = \sum \dot{m}_{\text{out}}^{k} h_{\text{out}}^{k} + \dot{Q}_{\text{loss}}^{k} \quad \text{(energy conservation)} \tag{18}$$

  Here, $\dot{m}_{\text{in}}^{k}$ and $\dot{m}_{\text{out}}^{k}$ represent the mass flow rates into and out of a control volume at time step k, respectively, and h_{in}^{k} and h_{out}^{k} denote the specific enthalpies of the incoming and outgoing streams at each k. Note that thermal losses to the ambient, $\dot{Q}_{\text{loss}}^{k}$, are not considered at the interconnection level between components but are included within the component models where relevant (e.g., for each TES).
- **Thermal stratification in TESs:** A monotonicity condition is imposed on each TES to ensure physically realistic temperature profiles. Specifically, the temperature in each stratified layer must not exceed the temperature in the layer directly above it, i.e., $T_{i+1}^{k} < T_{i}^{k}$. This condition enforces a top-down thermal gradient, ensuring that the upper layers remain hotter than the lower ones.
- **Thermal integration of STC output:** The thermal energy produced by the STC is directed toward the TES units (TES 1 and TES 2) and is stored in the layer with a temperature closest to the STC outlet temperature. This improves energy efficiency by reducing mixing. To model this behavior, a penalty term is included in the objective Function (16) that minimizes the squared temperature difference between the STC output and the TES layers. The penalty formulation is given as

$$\min \sum_{k=1}^{n} \left(\sum_{l=1}^{N} \alpha_{1,l}^{k} (T_{\text{TES},1,l}^{k} - T_{\text{STC,out}}^{k})^2 + \sum_{l=1}^{N} \alpha_{2,l}^{k} (T_{\text{TES},2,l}^{k} - T_{\text{STC,out}}^{k})^2 \right) \tag{19}$$

Here, $\alpha_{i,l}^k \in [0,1]$ are continuous weighting variables that represent the fraction of the STC flow directed to layer l of TES i at k, with the condition that

$$\sum_{l=1}^{N} \alpha_{i,l}^k = 1 \quad \text{for all } k,\ i \in \{1,2\} \tag{20}$$

This smooth layer-allocation approach avoids integer variables while still guiding the heat flow into the most suitable TES layer.

- **Heat demand constraint:** The process demand is supplied from the top node of TES 3 and must satisfy both temperature and mass flow rate constraints:

$$T_{\text{TES,3,out}}^k = T_{\text{demand}}, \quad \dot{m}_{\text{TES,3,out}}^k = \dot{m}_{\text{demand}} \tag{21}$$

with $T_{\text{demand}} \in [88\ ^\circ\text{C},\ 90\ ^\circ\text{C}]$ and $\dot{m}_{\text{demand}} \in [3000\ \text{L/h},\ 3500\ \text{L/h}]$.

It should also be noted that the underlying optimization problem uses two different time step sizes, reflecting the heterogeneous dynamics of the system components. The PV, STC, and TESs are updated every 15 min to capture short-term fluctuations. In contrast, the HP is resolved on an hourly basis to avoid frequent switching, due to its lower flexibility. This is done considering the following equality constraints for the inlet and outlet temperatures of the HP:

$$T_{\text{c,in}}^k = T_{\text{c,in}}^{k-1}, \quad T_{\text{h,in}}^k = T_{\text{h,in}}^{k-1}, \quad T_{\text{h,out}}^k = T_{\text{h,out}}^{k-1}, \quad \forall k \notin \{1,5,9,\dots\} \tag{22}$$

The 15 min step size is aligned with common electricity trading intervals, such as those in the intraday spot market [22]. The problem is formulated as a nonlinear problem, with all constraints being continuous and differentiable. Integer or binary variables are avoided through relaxation or continuous approximation.

The underlying optimization problem requires input data including solar irradiance for the PV and STC systems, the electricity price, the CO_2 emission factor, and the heat demand. The electricity price and emission factor were constant in this study and known in advance for the upcoming hours. Similarly, the required temperature and mass flow for the thermal energy supply process were also specified within fixed bounds and known in advance. Consequently, the only input that had to be forecast as part of the energy management process was the solar irradiance.

3. Hybrid Optimization Method

In this work, we address a constrained nonlinear nonconvex optimization problem, which becomes challenging as the problem's dimension and the number of constraints grow. To solve this optimization problem at each iteration of the RHA, we explore the novel hybrid BO–IPOPT recently proposed in the literature. We selected this approach in the current study because it has demonstrated superior performance compared to other state-of-the-art optimization algorithms in recent studies. The BO–IPOPT method is implemented as proposed in [18,19], without modifications. The purpose of this section is to provide a detailed summary of the algorithm to support the understanding of its integration in the RHA.

The BO–IPOPT is a hybrid optimization framework, first introduced in [18] and further enhanced in [19], that combines the global exploration of BO with the local refinement and constraint-handling strengths of the IPOPT, significantly reducing the risk of convergence to poor local minima. In this framework, BO is used to model the objective function and constraints via a surrogate model (GP model) and to guide the search toward promising regions in the solution space, without building a very detailed surrogate model. To handle

the constraints, BO follows the augmented LAGRANGIAN framework [28], which uses slack variables to transform inequalities into equality constraints and iteratively solve a sequence of simpler (unconstrained) problems. The AL framework is defined as follows:

$$u(\boldsymbol{x}, \boldsymbol{s}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}_g^\top (\boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{s}) + \boldsymbol{\lambda}_h^\top \boldsymbol{h}(\boldsymbol{x}) + \frac{1}{2\rho} \left[\sum_{w=1}^{q} (g_w(\boldsymbol{x}) + s_w)^2 + \sum_{r=1}^{p} h_r(\boldsymbol{x})^2 \right] \tag{23}$$

where $\rho > 0$ represents a penalty parameter controlling constraint violations and initialized based on an equation provided in [19], while $\boldsymbol{\lambda}_g \in \mathbb{R}_+^q$ and $\boldsymbol{\lambda}_h \in \mathbb{R}_+^p$ are the LAGRANGE multiplier vectors corresponding to inequality and equality constraints, respectively. The slack variable vector $\boldsymbol{s}$ is defined as

$$\boldsymbol{s} = \max\{\boldsymbol{0}, -\boldsymbol{\lambda}_g \rho - \boldsymbol{g}(\boldsymbol{x})\} \tag{24}$$

At each outer iteration K, the LAGRANGE multipliers, penalty parameter, and slack variables are iteratively updated, forming a dynamic process. The IPOPT then refines these regions identified through BO by performing local optimization, while efficiently handling complex constraints using an interior-point method. BO provides global search by suggesting new points to explore based on uncertainty and potential improvement, using the well-known and effective expected improvement (EI) acquisition function, while the IPOPT ensures that these points can be locally optimized with respect to defined constraints.

The BO–IPOPT algorithm operates through two key stages: an initialization phase followed by a loop that iterates up to a predefined number of outer iterations K.

In the initialization phase of Algorithm 1 (lines 1–4), an initial set of points is generated and evaluated using the augmented objective function. These evaluations are used to train a GP surrogate model using GPR. Unlike the traditional BO, which models the objective and constraints separately, this hybrid BO–IPOPT framework couples BO's global exploration capability with the IPOPT's local refinement and effective constraint handling. In this setup, the GP incorporates the parameters l and a, while the EI acquisition function uses the exploration factor ξ. Inputs such as $\boldsymbol{\lambda}_g$ and $\boldsymbol{\lambda}_h$ are required to compute the augmented objective function. For improved numerical stability and surrogate accuracy, all inputs and outputs to the GP are normalized to the $[0, 1]$ range.

The core loop of Algorithm 1 (lines 6–13) alternates between BO and the IPOPT. At each outer iteration, the acquisition function (EI) is evaluated to identify candidate points from a discrete set of points. Multiple candidates are selected at each iteration and passed to the IPOPT as starting points. The IPOPT then performs local optimization on the true objective function and constraints.

After completing all iterations, the algorithm returns the best solution found (line 14), which is the lowest objective function value in the evaluated dataset D—excluding the initial points D_0. This ensures that the result, $y_{\min}$, corresponds to a locally (and potentially globally) optimal solution of the investigated problem, while meeting all constraints.

By alternating between global exploration and local refinement at each iteration of the algorithm, the BO–IPOPT can efficiently solve high-dimensional optimization problems with numerous constraints, overcoming the challenges typically faced by traditional optimization methods in such settings. It should be mentioned that the hybrid method is parameter-free, since its parameters have been optimally set based on test cases considered in the previous work, and learning hyperparameters (common in BO approaches) during the optimization process has, thus, been avoided in our method. For further details about BO–IPOPT, the reader is referred to [18,19].

Algorithm 1 Hybrid Method BO–IPOPT

Input: Length scale l; regularization term α; exploration ξ; number of initialization points N_0; number of candidates N_c considered in EI; number of best candidates N_{bc}; Lagrange multiplier vectors $\boldsymbol{\lambda}_g$, $\boldsymbol{\lambda}_h$; number of outer iterations K

Generate an initial set of points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_0}\}$ in Ω
Evaluate $y_{i_0} = u(\boldsymbol{x}_{i_0}, \boldsymbol{s}_{i_0})$ for $i_0 = 1{:}N_0$
Let $D_0 = \{(\boldsymbol{x}_{i_0}, y_{i_0})\}_{i_0=1}^{N_0}$
Construct a GP model $\hat{u}_0$ from D_0
$z = N_0, j = 0$
while $j < K$ **do**
    Generate a new group of points $\{\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_{N_c}\}$ in Ω
    $\{\boldsymbol{x}_{z+1}, \ldots, \boldsymbol{x}_{z+N_{bc}}\} \in \arg\max_{\boldsymbol{x} \in D_j} \mathrm{EI}(\bar{\boldsymbol{x}}_i | \hat{u}_z)$ for $i = 1{:}N_c$
    Solve $[y_k^\star, \boldsymbol{x}_k^\star] = \mathrm{IPOPT}(\boldsymbol{x}_k)$ for $k = z+1{:}z+N_{bc}$
    $D_{j+1} = D_j \cup \{(\boldsymbol{x}_{z+1}^\star, y_{z+1}^\star), \ldots, (\boldsymbol{x}_{z+N_{bc}}^\star, y_{z+N_{bc}}^\star)\}$
    Update GP model $\hat{u}_{j+1}$ from D_{j+1}
    $z := z + N_{bc}, j := j + 1,$
end while
return $y_{\min} = \min\{y_j^\star\}_{j=1}^{KN_{bc}}$

4. Solar Irradiance Forecasting

Several forecasting techniques for solar irradiance have been proposed in the literature, generally falling into two broad categories: physics-based models and data-driven approaches. The former rely on atmospheric physics and numerical weather prediction tools, incorporating satellite data, cloud cover, and geographic information. While these models can provide relatively accurate forecasts, they require substantial computational resources and are not practical for real-time applications.

On the other hand, data-driven models—which include statistical and machine learning (ML) techniques—leverage historical solar irradiance measurements to identify patterns and trends in the data. These models are particularly advantageous, due to their lower computational complexity and adaptability. Previous studies have shown that while physics-based methods are more suited for long-term horizons, data-driven techniques offer superior performance for forecasts up to six hours ahead [29].

Given the need for fast and reliable solar irradiance predictions in the optimization framework used in this work, we adopted a data-driven approach. These models strike a balance between computational speed and forecasting accuracy, making them a practical choice for operational applications. The historical solar irradiance data used for training and validating the forecasting models were obtained from the GWS platform [30].

4.1. Data-Based Models

A wide range of data-driven modeling techniques have been explored in the literature for predicting solar irradiance, each offering distinct advantages depending on the complexity of the task and the available data [29,31–33]. Simple approaches such as the persistence model [29,31], which assumes that future solar irradiance will remain the same as the current value, are easy to implement and computationally efficient. However, their inability to account for rapid changes often results in limited predictive accuracy, especially under variable weather conditions.

Traditional statistical methods, including multiple linear regression (MLR) [29,32] and the autoregressive integrated moving average (ARIMA) [31,33], are frequently used to capture linear dependencies and temporal correlations within solar irradiance time series. In addition, multiple polynomial regression (MPR) [33] extends MLR by fitting higher-order polynomial terms, enabling the model to represent nonlinear trends in the data. While

these statistical models are effective for identifying basic patterns and seasonality, they often struggle with more complex dynamics.

To address these challenges, a variety of ML and deep learning (DL) algorithms have been adopted. Techniques such as random forest regression [29,31,32] and histogram-based gradient boosting regression (HGBR) [32] are particularly capable of modeling complex, nonlinear relationships by aggregating the predictions of multiple decision trees. The *k*-nearest neighbors (KNNs) algorithm [29], which bases predictions on the similarity to historical data points, is well-suited for capturing local patterns. Support vector regression (SVR) [29,31,32] is valued for its robustness in high-dimensional feature spaces and its ability to model nonlinear trends using kernel functions.

DL models have further advanced the field of solar irradiance forecasting. Artificial neural networks (ANNs) [29,32] are capable of learning nonlinear relationships from large datasets. More precisely, convolutional neural networks (CNNs) [31] efficiently extract spatial features from input data, while long short-term memory (LSTM) networks [31], a type of recurrent neural network, are designed to capture long-range temporal dependencies, making them highly effective for sequential prediction tasks.

Hybrid models, particularly those combining CNN and LSTM architectures, have demonstrated strong performance by leveraging both spatial and temporal information [31]. This combination has generally shown good results in weather forecasting applications, effectively capturing complex dependencies in meteorological data.

In this work, the focus was on the following models that balance accuracy with efficiency: MLR, MPR, ARIMA, KNN, SVR, HGBR, and a hybrid CNN–LSTM model.

4.2. Historical Data

This study utilized solar irradiance data obtained from the GWS platform, covering a 15-year period from 2010 to 2024, with a temporal resolution of 10 min. The dataset includes historical solar irradiance, sunshine duration, time without sun, precipitation, precipitation duration, precipitation height, air temperature at two meters, air pressure at station level, air pressure gradients between two locations at station level, and wind speed and wind direction at ten meters, as well as the relative humidity at two meters above ground level. In addition, a pseudo-parameter was derived from the GWS, taking the value of 1 for January, 2 for February, and increasing incrementally for the following months.

The analysis focused on two representative locations in Germany: Hamburg (north) and Augsburg (south). The forecasting models were specifically evaluated during two contrasting seasons—winter and summer of the year 2024.

As with many environmental datasets, the GWS data contain incorrect measurements marked by the placeholder value -999, along with occasional gaps. These were preprocessed and replaced, using one-dimensional monotonic cubic interpolation to ensure continuity for the model training and evaluation. This interpolation was also used to resample the solar data from a 10 min to a 15 min resolution, as required for the optimization problem presented in Section 2.

Identifying relevant input features is essential for improving the predictive accuracy of solar irradiance forecasts. In this study, the selection of predictive variables was based on two well-established statistical methods: the *p*-value method and the AKAIKE information criterion (AIC) [34].

The *p*-value method begins with a full model including all the possible features, gradually eliminating variables with the highest *p*-values until only statistically significant variables (i.e., those with *p*-values less than 0.05) remain. In addition, the AIC is used to

compare different possible models and determine which one is the best fit for the data. The formula for AIC is the following:

$$\mathrm{AIC} = 2K - 2\ln(L) \tag{25}$$

where K is the number of predictors and L is the maximum likelihood estimate. A lower AIC value indicates a more optimal balance between model complexity and goodness of fit.

Based on these selection criteria and taking into account the seasonal variability of the variables, the most relevant predictors identified are the solar irradiance from the previous four time steps, as well as solar irradiance from one day and one year ago, the sunshine duration from one day ago, and the temperature and pressure from one day ago. The resulting solar irradiance is expressed by

$$I_t = f(I_{t-1}, I_{t-2}, I_{t-3}, I_{t-4}, I_{t-\mathrm{day}}, I_{t-\mathrm{year}}, SD_{t-\mathrm{day}}, T_{t-\mathrm{day}}, p_{t-\mathrm{day}}) \tag{26}$$

where I denotes the solar irradiance, SD the sunshine duration, T the temperature, and p the pressure.

These input features are considered across all the models discussed in Section 4.1, except for the ARIMA model, which only incorporates the solar irradiance at previous time steps. MPR also incorporates interaction terms between lagged irradiance values, so that the resulting model for the second-degree MPR is given by

$$\begin{aligned} I_t = f(&I_{t-1}, I_{t-3}, I_{t-4}, I_{t-\mathrm{day}}, I_{t-\mathrm{year}}, SD_{t-\mathrm{day}}, T_{t-\mathrm{day}}, p_{t-\mathrm{day}}, I_{t-1}^2, I_{t-1}I_{t-3}, \\ &I_{t-1}I_{t-4}, I_{t-1}I_{t-\mathrm{day}}, I_{t-1}I_{t-\mathrm{year}}, I_{t-3}^2, I_{t-3}I_{t-4}, I_{t-3}I_{t-\mathrm{day}}, I_{t-3}I_{t-\mathrm{year}}, \\ &I_{t-4}I_{t-\mathrm{day}}, I_{t-4}I_{t-\mathrm{year}}, I_{t-\mathrm{day}}^2, I_{t-\mathrm{day}}I_{t-\mathrm{year}}, I_{t-\mathrm{year}}^2) \end{aligned} \tag{27}$$

In addition, the forecasting process checks whether it is day or night at each prediction. When it is night, based on sunrise and sunset times, the solar irradiance is set to zero to reflect the absence of solar input during these hours.

4.3. Feature Scaling

ML and DL models, like KNN, SVR, and ANN, often require feature scaling to perform effectively. This preprocessing step transforms input features so they share a similar numerical range, ensuring that no single feature disproportionately influences the model's behavior. By applying feature scaling, the training process becomes more stable and efficient. The two most widely used techniques are standardization and normalization, defined by the following equations [35]:

$$x_i' = \frac{x_i - \bar{x}}{s} \quad \text{(standardization)} \tag{28}$$

$$x_i' = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \quad \text{(normalization)} \tag{29}$$

where $\bar{x}$ and s are the mean value and the standard deviation, respectively.

4.4. Performance Evaluation Criterion

As discussed in Section 4.1, both linear and non-linear forecasting models introduce errors due to statistical approximations. To assess the accuracy of these models, one commonly used performance metric is applied: mean absolute error (MAE). This metric is defined as follows [36]:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{30}$$

where y_i and $\hat{y}_i$ denote the measured and predicted values, respectively, and N is the number of data points considered in the performance evaluation.

4.5. Cross-Validation (CV)

CV is a data resampling technique used to evaluate the generalization performance of predictive models and to mitigate the risk of overfitting [37]. Given the frequent occurrence of overfitting in ML and DL applications, this study applied CV to the models introduced earlier to ensure more robust and reliable performance.

Among the various CV strategies, k-fold CV is one of the most widely used. In this method, the dataset is divided into k equally sized folds. The model is then trained on $k-1$ folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the testing set once. The overall model performance is reported as the average of the evaluation metrics across all k iterations.

However, applying CV to time series data presents specific challenges. Unlike conventional datasets, it is inappropriate to randomly reorder time series samples, as this could result in future values being used to predict past observations—an unrealistic and invalid approach. To address this, a blocked CV method was adopted, based on the TIMESERIESSPLIT functionality provided by the SCIKIT-LEARN library [38], and further refined according to the approach described in [39].

As illustrated in Figure 2, two types of margins were introduced to improve the validity of the evaluation. The first margin was inserted between the training and testing sets to avoid data leakage and ensure that no samples were re-used. The second margin was placed between the folds to prevent the model from learning temporal dependencies between sequential iterations, thus reducing the risk of overfitting across splits.

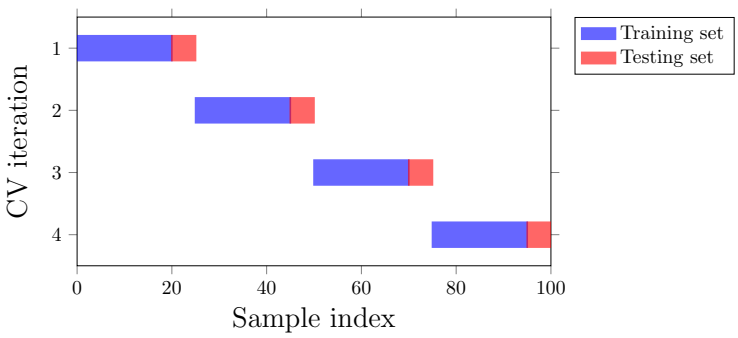

Figure 2. Schematic structure of blocked CV for time series split.

4.6. Multi-Step-Ahead Forecasting Strategy

Forecasting multiple steps ahead is significantly more challenging than single-step forecasting, due to the accumulation of prediction errors, decreased accuracy, and increasing uncertainty [40]. While several strategies for multi-step forecasting have been proposed—five of which are outlined in [41]—this work focused exclusively on the recursive strategy, a widely used and well-established method in the literature, defined by

$$\begin{aligned} \hat{I}_t &= f(I_{t-1}, I_{t-2}, I_{t-3}, I_{t-4}, I_{t-\text{day}}, I_{t-\text{year}}, SD_{t-\text{day}}, T_{t-\text{day}}, p_{t-\text{day}}) \\ \hat{I}_{t+1} &= f(\hat{I}_t, I_{t-1}, I_{t-2}, I_{t-3}, I_{t+1-\text{day}}, I_{t+1-\text{year}}, SD_{t+1-\text{day}}, T_{t+1-\text{day}}, p_{t+1-\text{day}}) \end{aligned} \tag{31}$$

To forecast H future steps, the model's prediction at each step is fed back as an input for the next step. This iterative process continues until the full forecasting horizon is reached.

4.7. Results

This section analyzes the forecasting performance of the models introduced in Section 4.1. The ARIMA model considered follows the (0,2,2) configuration, selected based on insights from the autocorrelation and partial autocorrelation plots. For the ML and DL approaches, hyperparameter optimization was carried out to ensure robust predictive accuracy. To mitigate overfitting, we adopted the blocked CV. All the forecasting models were implemented using Python 3.8. Specifically, the CNN–LSTM model leveraged the TENSORFLOW library, ARIMA was implemented through the STATSMODELS package, and the remaining ML models were built using SCIKIT-LEARN. All computations were performed with an Intel(R) Xeon(R) Gold 5220 CPU running at 2.20 GHz.

As expected, the forecasting error increased with the number of predicted time steps, as illustrated in Figure 3. For one-step-ahead predictions, most of the models demonstrated comparable accuracy, except for CNN–LSTM and ARIMA, which showed slightly higher errors compared to the other models. As the prediction horizon extended, the performance of ARIMA degraded notably, with the MAE increasing from 0.26 to 0.46 during the winter week (≈79% increase) and from 0.19 to 0.37 during the summer week (≈91% increase), indicating a significant drop in predictive accuracy over time. This was due to the model's reliance on past errors (the moving average component), which were not available for future time steps and were assumed to be zero—an assumption that proved unreliable for multi-step forecasting. Consequently, ARIMA is particularly well-suited for single-step forecasts but performs poorly in longer horizons.

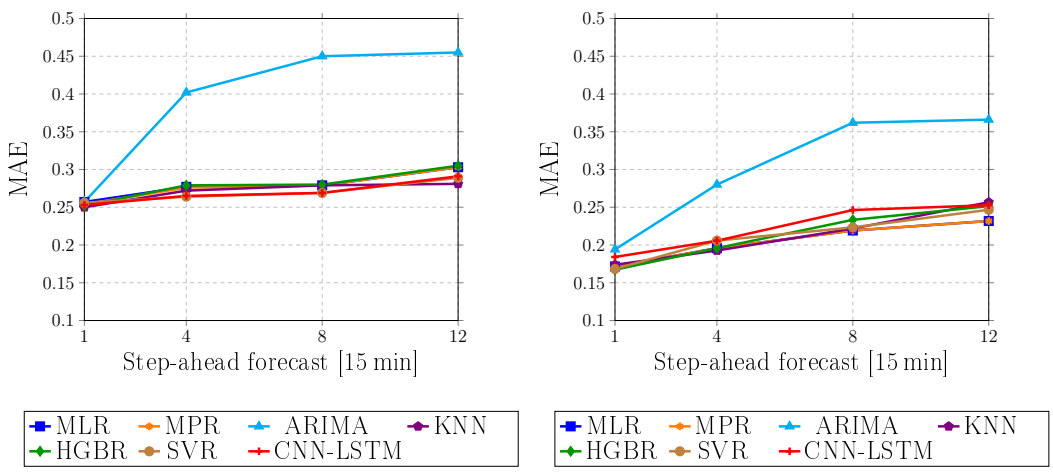

Figure 3. Results of solar irradiance forecast using MAE comparison for the tested methods with respect to different multi-step-ahead horizons, for Hamburg in winter (**left**) and Augsburg in summer 2024 (**right**).

On the other hand, MLR, MPR, KNN, SVR, HGBR, and CNN–LSTM showed a more stable error progression over multiple steps, with relatively moderate increases in MAE, as well as a similar level of accuracy over the forecast horizon. On average, their MAE increased from 0.25 to 0.30 during the winter week (≈20% increase), and from 0.17 to 0.25 during the summer week (≈47% increase). However, the models tended to produce more accurate predictions in August (summer) than in January (winter), showing up to a 30.8% decrease in error, on average (first forecasting step). This seasonal difference can be explained by the higher and more stable solar irradiance levels in summer, which provide a clearer input signal and reduce the relative impact of forecast noise. In contrast, irradiance during winter is lower and more variable, making it harder for the models to identify reliable patterns.

In terms of computational requirements, the training time of each model varied significantly. ARIMA, SVR, and CNN–LSTM required more computational resources than the other models. Specifically, CNN–LSTM needed around 711 s to train, while ARIMA and SVR required approximately 60,321 and 613 s, respectively. In contrast, the simpler models such as MLR, MPR, HGBR, and KNN completed the training process in under 6 s, with MLR being the fastest (0.10 s).

Overall, MLR, HGBR, and KNN offer effective choices for solar irradiance forecasting, balancing accuracy and computational efficiency. In future work, integrating filtering techniques could help enhance the model performance further.

5. Energy Management

This section describes in detail the methodology applied for the energy management of the considered power-to-heat system as well as the results of it, integrating the novel method "BO–IPOPT".

5.1. Methodology

The primary objective of this study was to operate the industrial energy system described in Section 2 in a cost- and emission-efficient manner by minimizing electricity costs from the grid and CO_2 emissions, while considering the uncertainties associated with fluctuating solar energy. The role of the energy management within the broader framework of real-time optimization is illustrated on the left of Figure 4. This framework is typically structured into three hierarchical levels: system-level energy management, component-level control, and physical process operation.

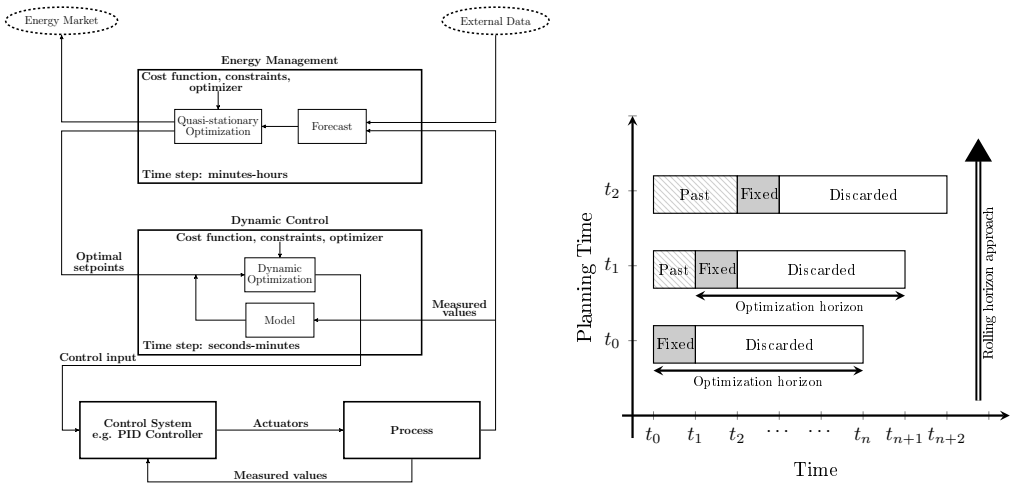

Figure 4. Detailed block structure of the different time levels for real-time optimization of a system (**left**) and visualization of the RHA (**right**).

At the top level (system level), which was the focus of this work, predictive optimization is carried out on a timescale ranging from minutes to hours. This level involves solving a multi-period optimization problem to determine cost- and emission-minimizing setpoint trajectories for system variables such as temperature, mass flow rate, and power. These trajectories are subsequently passed to the second level (component level), where dynamic controllers ensure stable system behavior by managing process dynamics at finer time steps (seconds to minutes). The third level focuses on the real-time execution of control signals by physical actuators, such as valves and pumps. In this study, only the system-level optimization was considered; component-level dynamics were excluded. Additionally, the system-level optimization was directly coupled with market operations—e.g., participating in electricity trading.

As outlined in Section 1, this study employed the RHA, shown on the right of Figure 4, which provides a structured and adaptive method for handling uncertainties in energy management. The RHA repeatedly solves an optimization problem over a moving time window. After each iteration, only the first control action is implemented, while the rest of the computed trajectory is discarded. The horizon then moves forward to incorporate new real data, enabling ongoing refinement of operational decisions.

For this purpose, the open-source CoDeOpT tool can be used, which was first introduced in [20] and further enhanced in the current work to support the energy management of systems like the one presented in this study. As seen in Figure 5, CoDeOpT connects forecasting, system modeling, optimization, and data exchange with the real plant in a modular architecture designed for industrial applications. It utilizes weather data from the Gws platform via an API and processes day-ahead electricity price data from the Smard platform [42]. At each RHA iteration, CoDeOpT computes optimal setpoints and provides them to the real plant through the interface "Sinnogenes Middleware", while simultaneously receiving actual plant values, enabling the system to adapt its operation based on actual process conditions and deviations from expected behavior. However, this study did not consider real-time measurements. In addition, the first step of the workflow, performed offline, is to determine an optimal system design and structure, which serves as the foundation for the subsequent energy management and real-time operation phases.

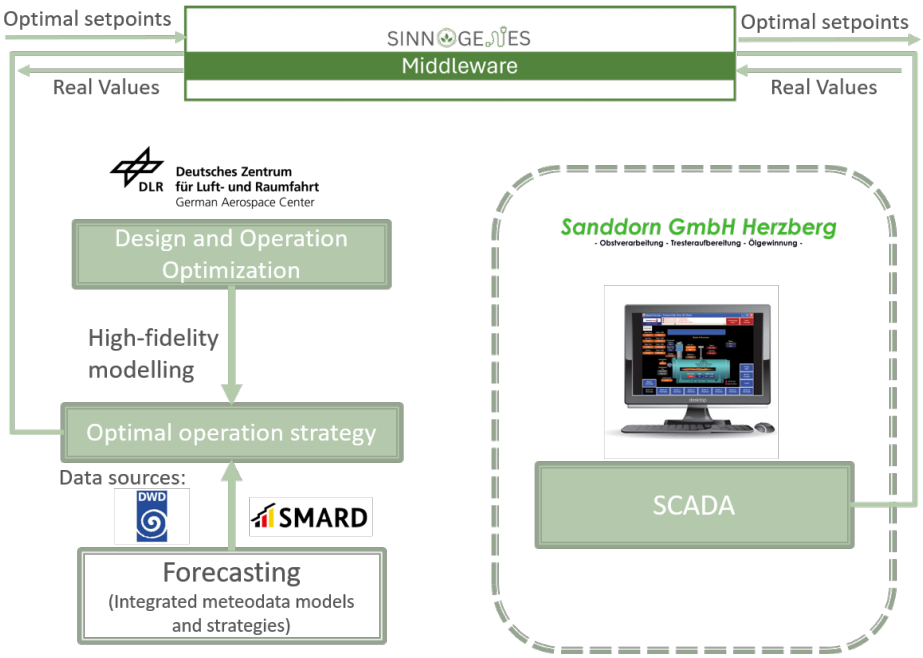

Figure 5. Visualization of CoDeOpT and the real plant interface "Sinnogenes Middleware".

In this work, solar irradiance was considered as the fluctuating input variable, with data selected for one representative week in January and another in August at a single location in Germany—capturing periods of low and high solar activity, respectively (see Figure 6). The solar data were obtained from the Gws platform. The electricity price and the CO_2 emission factor associated with grid electricity were assumed to be fixed and known in advance (see Section 2). Since the solar data were originally available at a 10 min resolution, they were resampled using monotonic cubic interpolation to align with the 15 min resolution required by the energy management framework.

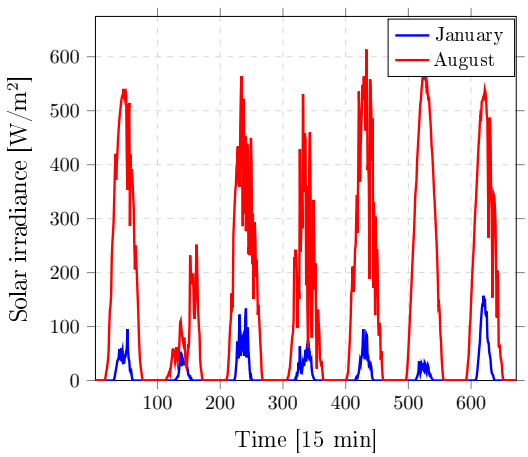

Figure 6. Visualization of one-week-scenario input data: solar irradiance for Herzberg (Elster), Germany in winter and summer 2024.

5.2. Results

The effectiveness of energy management in our system is influenced by several key factors, including the choice of optimization algorithm, the allowed computation time per iteration of the RHA, the forecasting model used for solar data, and the length of the optimization horizon. These parameters are summarized in Table 3. For this section, we systematically evaluated the impact of varying each of these parameters on the system performance. First, we evaluated the performance of the new hybrid method BO–IPOPT, in comparison with the widely used stochastic optimizers "GA" and "PSO". GA is inspired by the process of natural selection, where a population of candidate solutions evolves over generations through operations such as selection, crossover, and mutation. PSO, on the other hand, mimics the social behavior of swarms, where candidate solutions, called particles, adjust their positions in the search space based on both individual and collective experience. Both methods are widely used in nonlinear and black-box optimization problems and are commonly applied in nonlinear RHA-based energy management tasks.

Table 3. Energy management parameters.

Parameter	Description
Optimizer	GA, PSO, BO–IPOPT
CPU running time	20 s, 30 s, 40 s, 50 s
Forecasting model	MLR, HGBR, KNN, CNN–LSTM
Optimization horizon length	8 steps, 12 steps, 16 steps, 20 steps, 24 steps

All the methods were implemented in PYTHON 3.8. The IPOPT was accessed via the PYOMO optimization modeling framework, using its default configuration. GA and PSO adopted default parameter settings from [43,44]. The BO–IPOPT was configured according to recommendations from [19], with the number of best-performing candidates per outer iteration set to four—corresponding to the number of available CPU cores. The parallelization of the IPOPT runs in the BO–IPOPT was achieved via the PYTHON's library "MULTIPROCESSING" to enhance computational efficiency. However, to ensure fair comparison with GA and PSO, the BO component in the hybrid method was not parallelized.

All the experiments were conducted on a machine equipped with an Intel(R) Core(TM) i7-8665U CPU. For GA, PSO, and the BO–IPOPT, the optimizer was allowed to run for a fixed time duration (running time) at each RHA step to search for the best possible solution (i.e., the optimal control trajectories for the system). It is worth noting that all the optimization methods considered in this study are based on a certain degree of

randomness. For this reason, we repeated each numerical experiment 10 times to average out the stochastic nature of the optimizers.

Moreover, we used a baseline optimization horizon of 8 time steps, equivalent to 2 h at a 15 min resolution, resulting in an optimization problem of 1064 dimensions at each RHA iteration. This choice balanced the problem complexity with the reliability of the solar forecasting models. However, the impact of varying the optimization horizon is analyzed later in this work.

When comparing optimization results across different settings, the global minimum ideally serves as the reference solution. However, since the global minimum was unknown, we used the best solution found by the BO–IPOPT, which ran with a 5 min time limit per RHA iteration, as a benchmark for performance comparison.

First, we evaluated the impact of different optimizers integrated into the RHA framework. To ensure a fair comparison, all the numerical experiments were performed under ideal conditions, i.e., without uncertainties in the input data. Furthermore, the optimizer's running time was fixed to 50 s per RHA iteration, a value chosen to balance the need for real-time control signal exchange and system response with the computational complexity and high dimensionality of the underlying optimization problem. The optimization outcomes are summarized in Figure 7, which presents the operating costs and CO_2 emissions for a representative winter and summer week. The results are visualized as box plots, showing the distribution of the accumulated objective values over the repeated experiments for each optimizer.

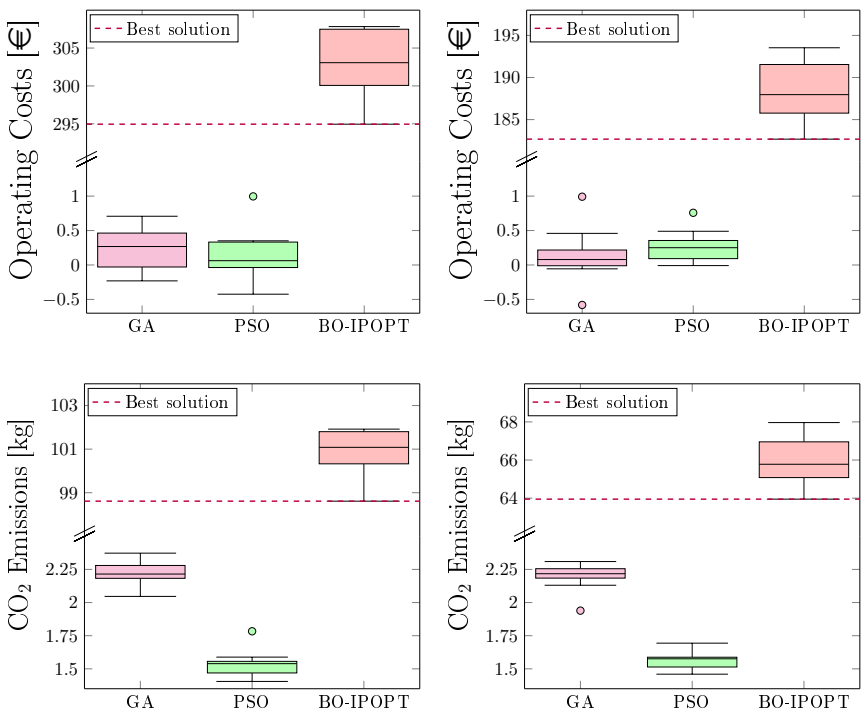

Figure 7. Comparison of operating costs (**top**) and CO_2 emissions (**bottom**) over 10 trials for different optimizers in winter (**left**) and summer (**right**) with an optimizer running time of 50 s per RHA iteration, an eight-step optimization horizon, and assuming that the input data are known. The circles in the box plots represent outliers.

In all cases, the novel hybrid method the BO–IPOPT clearly outperformed the stochastic solvers "GA" and "PSO", in terms of both costs and emissions. In the winter week, the BO–IPOPT achieved an average operating cost, measured in terms of the median, of EUR 303.07, with a relative error of 2.75% compared to the best known solution (EUR 294.95). In contrast, GA and PSO yielded significantly worse results—EUR 0.27 and EUR 0.06, cor-

responding to relative errors of 99.9% and 100%, respectively. It is important to note that the lower values observed in GA and PSO are misleading, as they resulted from violated constraints—penalties embedded in the objective function—with constraint violation values in the order of 10^{10}. These solvers struggled to satisfy the constraints within the short time limit. In contrast, the BO–IPOPT combines the global search capability of BO with the fast local convergence of IPOPT, enabling both constraint satisfaction and superior optimization performance.

This discrepancy can be attributed to the structural limitations of GA and PSO in constrained, high-dimensional, nonlinear optimization problems like the one considered here. These algorithms rely on population-based search strategies and stochastic operators, which often require significantly more iterations to converge to feasible and optimal regions. However, due to the tight CPU time limits imposed in the real-time framework, they are unable to sufficiently explore and refine their solutions. In contrast, the BO–IPOPT leverages the global exploration ability of BO to identify promising regions, followed by the rapid, gradient-based convergence of the IPOPT for local refinement. This hybridization ensures both constraint satisfaction and improved optimization performance, even within strict time constraints.

In the summer week, where higher solar irradiance enabled greater use of PV and STC systems, the BO–IPOPT achieved a lower operating cost of EUR 187.96, with a relative error of 2.9% compared to the best solution (EUR 182.68). GA and PSO again performed poorly, with objective values of EUR 0.08 and EUR 0.25, corresponding to relative errors of 100% and 99.9%, respectively.

The same pattern held for the CO_2 emissions. In winter, the BO–IPOPT reached 101.08 kg, with a relative error of 2.5% compared to the benchmark value of 98.61 kg. GA and PSO showed much lower emissions—2.21 kg and 1.54 kg—translating to relative errors of 97.8% and 98.4%, respectively. In summer, the BO–IPOPT led to 65.78 kg CO_2 with a relative error of 2.9% compared to the best solution (63.95 kg), while GA and PSO resulted in 2.22 kg and 1.58 kg CO_2, i.e., relative errors of 96.5% and 97.5%, respectively. The performance of each optimizer is summarized in Table 4. Overall, the BO–IPOPT outperformed the other two stochastic optimizers across all four scenarios, and it was, thus, used exclusively in the following investigations of this study.

Table 4. Summary of optimizer performance over 10 trials with 50 s running time per RHA iteration. Each cell contains the winter value followed by the summer value, separated by a slash (winter/summer).

Optimizer	Mean Cost [€]	Best Cost [€]	Mean CO_2 [kg]	Best CO_2 [kg]
GA	0.27/0.08		2.21/2.22	
PSO	0.06/0.25	294.95/182.68	1.54/1.58	98.61/63.95
BO–IPOPT	303.07/187.96		101.08/65.78	

In many real-time applications, it is crucial to keep the optimizer's CPU running time as short as possible to ensure that the optimized operational strategy is generated in time to react effectively to rapidly changing system conditions. Figure 8 presents the optimization results of the BO–IPOPT across different CPU running times—20 s, 30 s, 40 s, and 50 s per RHA iteration—for both operating costs and CO_2 emissions under winter (left) and summer (right) conditions. As expected, extending the CPU time generally improved the optimization quality. For the winter dataset, the average operating costs decreased by approximately 0.7% from 20 s to 50 s, while the CO_2 emissions reduced by around 1.0%. Similarly, under summer conditions, the operating costs dropped by about 1.8%, and the emissions fell by 2.2% over the same interval. Although the average improvements

in objective values were relatively modest, a more pronounced benefit of the increased running time was the narrowing of the box plots—representing the spread of outcomes—which became particularly clear at 50 s. This indicates a significant gain in the consistency and reliability of the optimization results. Notably, even at shorter running times (20 s and 30 s), the optimizer was able to identify near-optimal solutions, although with higher variability. These findings confirm that longer running times enhance both solution quality and robustness. While running times beyond 50 s can offer advantages, in terms of improved optimization quality—such as greater robustness and reductions in operating costs and emissions—they were less suitable for the current study, due to the practical limitations. In this study, we fixed the CPU time per RHA iteration to 50 s, as it provided a good trade-off between solution accuracy and computational speed. This choice enabled efficient execution of the further investigations, where longer running times would significantly increase the overall CPU time. Even with this 50 s limit, the optimizer demonstrated reliable performance, allowing us to draw meaningful conclusions about the impact of other factors—uncertainties in the solar data and the length of the optimization horizon—on the optimization outcomes. In future work, the optimization framework will run on a LINUX server interfaced directly with the real plant. This setup is expected to support either extended running times or, more effectively, a higher number of evaluations within the same running time window, enabling faster and more reliable real-time optimization.

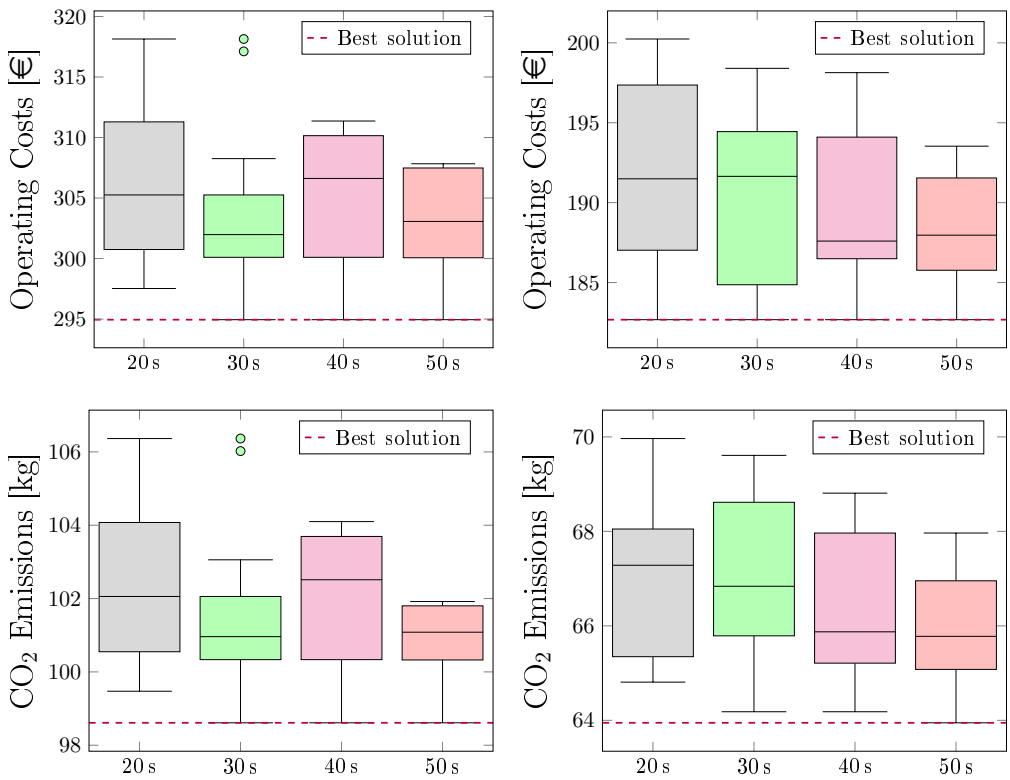

Figure 8. Comparison of operating costs (**top**) and CO_2 emissions (**bottom**) over 10 trials for different running times of the BO–IPOPT in winter (**left**) and summer (**right**) with an eight-step optimization horizon and assuming that the input data are known.

As previously discussed, accounting for uncertainties in the input data is essential to reflect more realistic operating conditions. In this context, we integrated the solar irradiance forecasting models into the RHA and evaluated their impact on the system performance. The models tested included MLR, HGBR, and KNN using a recursive forecasting strategy—chosen for their good balance between predictive accuracy and computational efficiency, with training times under 5 s (see Section 4.7). Additionally, we evaluated a CNN–LSTM

model, which, due to its higher computational cost, was trained only once at the beginning of the energy management process and not retrained at each RHA iteration like the other models. This model was selected for this section since it demonstrated good forecasting performance in Section 4.7.

Figure 9 presents the results for a fixed optimizer running time of 50 s and an eight-step optimization horizon per RHA iteration. For both the winter (left) and the summer (right) weeks, incorporating forecasted solar data using the MLR, HGBR, and KNN models led to noticeable increases in operating costs and CO_2 emissions compared to the idealized scenario with perfect (known) solar data. The maximum deviations reached, on average, approximately 5.8% in costs and 4.4% in emissions. These increases were primarily due to the systematic underestimation of solar irradiance by the forecasting models, which reduced the utilization of the available solar energy and increased the dependency on grid electricity.

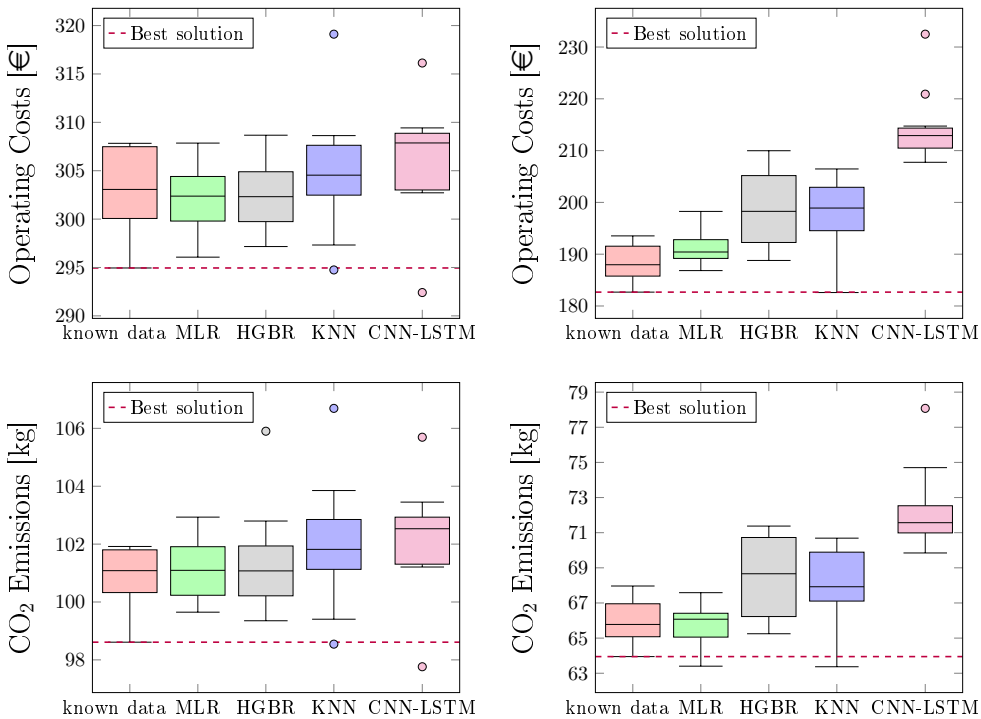

Figure 9. Comparison of operating costs (**top**) and CO_2 emissions (**bottom**) over 10 trials for different forecasting models for the solar data using the BO–IPOPT with a running time of 50 s and an eight-step optimization horizon in winter (**left**) and summer (**right**).

Among the tested models, MLR consistently delivered strong performance with minimal deviation from the known data baseline—remaining within 1% across all cases—while requiring very low training effort. In contrast, the CNN–LSTM model showed higher variance and average values (up to 13.3% in costs and 8.8% in emissions), likely due to its lack of retraining during the RHA iterations, reducing its ability to respond to changing data trends. Overall, these findings underline the suitability of simple models such as MLR for real-time energy management tasks, particularly in scenarios where computational efficiency and high accuracy are required. For this reason, MLR was used in the following analysis.

The final parameter investigated in this work was the length of the optimization horizon, which plays a critical role in defining the operational strategy within the RHA-based energy management framework. The horizon length affects the performance in two key ways. First, longer horizons enable the optimizer to better account for future system conditions—such as fluctuations in renewable energy availability and electricity

prices—leading to more informed and strategic operation, especially in the management of energy storage systems. Second, extending the horizon increases the dimensionality and complexity of the optimization problem, making it more computationally demanding and potentially more sensitive to forecast uncertainties. In practical applications, this trade-off between planning depth and computational feasibility must be carefully managed. For the numerical experiments presented in this study, we evaluated optimization horizons of 8, 12, 16, 20, and 24 steps, corresponding to 2, 3, 4, 5, and 6 h (based on a 15 min resolution), with a fixed CPU running time of 50 s per RHA iteration.

Figure 10 shows that increasing the optimization horizon at each RHA iteration led to reductions in both operating costs and CO_2 emissions, under both winter and summer conditions. This was because the longer horizons allowed the optimizer to better predict future system behaviors—especially the availability of renewable energy sources such as PV and STC—and make more informed, forward-looking operational decisions. Additionally, the best solution achieved improved (i.e., the objective value decreased) as the horizon increased. However, as observed for the 24-step horizon, the spread in the results increased. This suggests that the optimization problem becomes more complex with increasing horizon length, and the fixed 50 s running time may no longer be sufficient to consistently find high-quality solutions. Horizons longer than 24 steps were not tested, as the associated increase in decision variables and constraints would have exceeded the computational capacity within the 50 s limit.

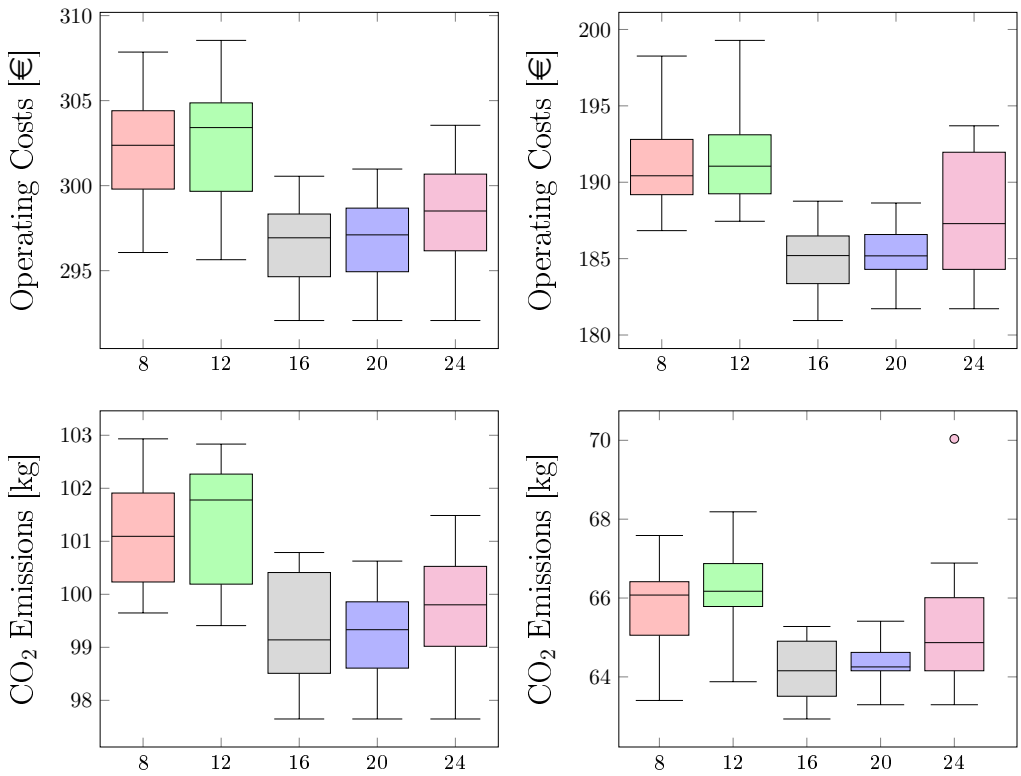

Figure 10. Comparison of operating costs (**top**) and CO_2 emissions (**bottom**) over 10 trials for different optimization horizons (in steps) using the BO–IPOPT with a running time of 50 s and the MLR method for the solar data forecast in winter (**left**) and summer (**right**).

Figure 11 compares the influence of forecast uncertainty by contrasting the optimization results using forecasted solar data (MLR) with those using perfect, known data. Even for the longest tested horizon of 24 steps, the increase in operating costs and emissions due to forecast errors was limited to approximately 1.24% and 0.7%, respectively. These differences are comparable to those observed for shorter horizons (see Figure 9), indicating

that the BO–IPOPT maintains robust performance in the presence of solar prediction uncertainties. This also suggests that the employed MLR forecast model delivers sufficiently reliable input for effective real-time control.

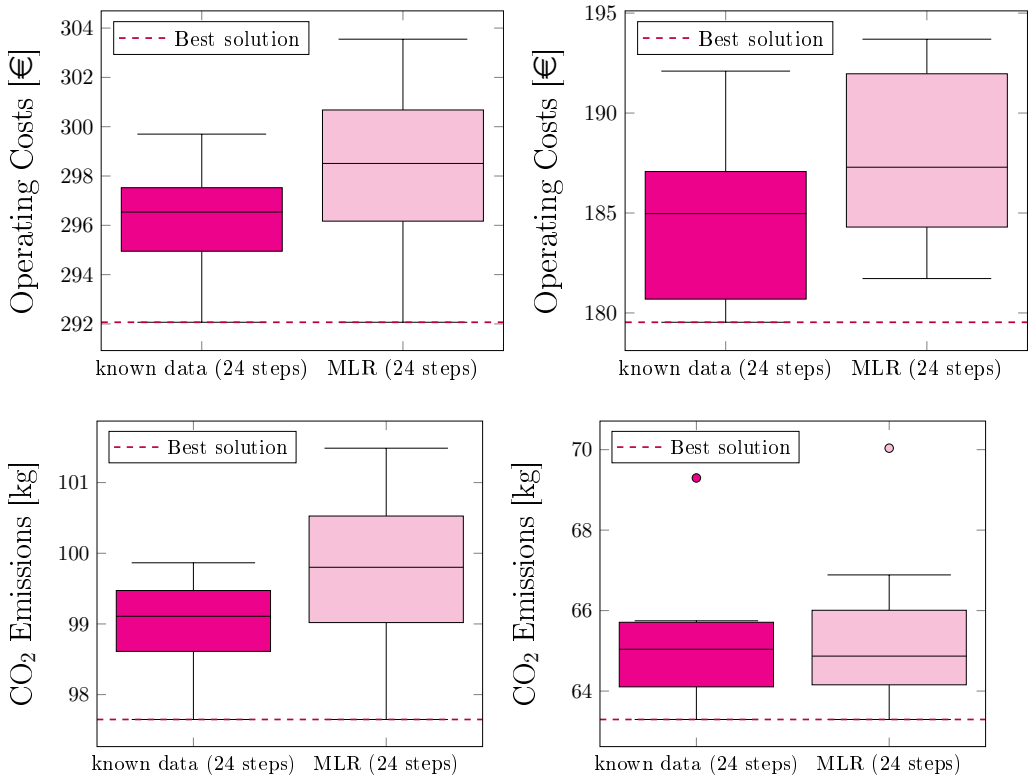

Figure 11. Comparison of operating costs (**top**) and CO_2 emissions (**bottom**) over 10 trials using the BO–IPOPT with a running time of 50 s, a 24-step optimization horizon, and the MLR method for the solar data forecast in winter (**left**) and summer (**right**). The results compare the case without forecast uncertainties to the case with uncertainties.

6. Conclusions

This work presents a detailed modeling and analysis of a novel industrial energy system, which integrates an HP, STC, multiple TESs, and PV generation to efficiently meet the process heat demand. We evaluated the performance of the hybrid optimization approach "BO–IPOPT" for the real-time energy management of this system—an approach not previously used in this context. Specifically, we analyzed the method's accuracy and robustness in comparison with established state-of-the-art optimization techniques. Furthermore, we examined how key factors—such as the optimizer's CPU running time per RHA iteration, uncertainties in solar irradiance forecasts from different prediction models, and variations in the optimization horizon length—affect the system's operational decisions and overall efficiency.

The results show that the the BO–IPOPT significantly outperformed the popular stochastic optimization methods "GA" and "PSO", achieving up to 97.25%-better objective function values at the same CPU time. While GA and PSO struggled to satisfy the numerous constraints within the required CPU running time at each RHA iteration, the BO–IPOPT effectively balanced global exploration and fast local convergence, enabling it to achieve superior optimization performance and constraint satisfaction.

Regarding CPU time, the results indicate that increasing the optimizer's running time per RHA iteration not only improves the solution quality but also enhances the robustness, as reflected by a narrower spread in the objective values. For instance, extending the CPU

time from 20 s to 50 s led to reductions in average operating costs by up to 1.8% and CO_2 emissions by up to 2.2%, depending on the season. Nevertheless, even at shorter running times, the BO–IPOPT is able to deliver near-optimal solutions with reliable constraint satisfaction. A fixed running time of 50 s was chosen for this study, as it provides a practical balance between accuracy and computational efficiency, enabling consistent performance while keeping the overall computation time manageable for further investigations.

We also studied uncertainties in the solar irradiance data by evaluating the performance of different data-driven forecasting models. Among them, the simple MLR model showed a favorable balance between prediction accuracy and computational efficiency—particularly due to its very short training time—making it a practical and effective choice for real-time energy management applications. Despite the presence of input uncertainties, the optimization results with MLR remained within 1% of those obtained under ideal conditions where all inputs are assumed to be known, highlighting its robustness. Although the accuracy was already high, future work could explore filtering techniques applied to the data to further enhance the forecasting model performance.

With respect to the optimization horizon, the results demonstrate that longer horizons generally lead to improved performance by allowing better prediction of renewable energy availability and system dynamics. This supports more strategic operation, especially in the use of storages. However, increasing the horizon also enlarges the problem size, making it more computationally intensive and potentially exceeding the available CPU time, leading to greater variability in the results. In this study, horizons of 16 and 20 steps offered a good trade-off between predictive planning and computational feasibility under a fixed 50 s running time. While these findings are specific to the energy system considered in this work, the observed balance between horizon length, solution quality, and computational effort is likely applicable to systems of similar scale and complexity. For larger or more complex systems, maintaining real-time feasibility may require extending CPU time or shortening the horizon—or, alternatively, leveraging more powerful server infrastructure to enable a higher number of evaluations within the same running time.

While the results of this study demonstrate the strong technical performance of the BO–IPOPT method within the energy management environment, several limitations must be acknowledged. First, the findings are based on a case-specific model and have not yet been validated in real plant operation. As a next step, we aim to implement the proposed approach in a Hardware-in-the-Loop setup or directly within the actual plant infrastructure. In such a setup, setpoints generated by the BO–IPOPT will be transmitted to system components, which then attempt to reach the desired states at each control interval. Real-time feedback from sensors will allow dynamic model updating and assessment of closed-loop behavior under realistic operating conditions. To support this, the optimization tool will run on a LINUX-based server interfaced directly with the real plant, allowing smooth integration and online execution. This setup will enable us to validate the underlying models used in this study, adapt them if needed, and further assess the practical applicability and effectiveness of the overall methodology in a real-world operational environment.

Additionally, although the hybrid optimization approach scales well for the current system size, its computational performance for significantly larger or more complex systems remains to be tested. Moreover, the analysis focused primarily on uncertainties in solar forecasts, while other critical sources of variability—such as demand profiles and electricity price volatility—were not fully explored. Future research should, therefore, investigate the method's robustness under broader uncertainty scenarios and test its performance on different industrial configurations to verify generalizability.

Finally, while the emphasis was placed on the algorithmic performance, the overall goal remains to improve the economic and environmental sustainability of industrial energy

systems. To enhance real-world applicability and sustainability, future work could incorporate time-varying electricity prices and flexible demand profiles into the optimization framework. The current assumption of a constant, inflexible demand over a 24 h horizon may not reflect operational flexibilities available in many industrial settings. Integrating demand-side flexibility would enable dynamic adjustments to energy consumption based on price signals and renewable generation forecasts, thereby reducing operating costs and CO_2 emissions and advancing the decarbonization goals.

Author Contributions: L.K.: conceptualization, methodology, validation, formal analysis, investigation, resources, data curation, software, writing—original draft, writing—review and editing, visualization; R.K.: conceptualization, writing—review and editing; M.I.R.S.: project administration, supervision, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union's Horizon Europe project "SINNOGENES" (storage innovations for green energy systems), under Grant Agreement No. 101096992.

Data Availability Statement: Restrictions apply to the availability of these data. The data were obtained from Martin Brylka and are available from the authors with the permission of Martin Brylka.

Acknowledgments: We would like to thank Martin Brylka for providing data and for continuous support.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

The following symbols and abbreviations are used in this manuscript:

Latin Symbols:

A	collector area [m^2]
c_p	specific heat capacity [J/(kg K)]
D	set of points [various units]
E	energy [J]
f	objective function [various units]
$\boldsymbol{g}$	set of inequality constraints [various units]
h	specific enthalpy [J/kg]
$\boldsymbol{h}$	set of equality constraints [various units]
I	solar irradiance [W/m^2]
k	thermal conductivity [W/(m K)]
K	incidence correction factor, number of outer iterations [-]
l	length scale [various]
$\dot{m}$	mass flow rate [kg/s]
N	number of vertical layers, number of points [-]
p	pressure [Pa]
P	power [W]
$\dot{Q}$	heat transfer rate [W]
$\boldsymbol{s}$	slack variable vector [various units]
SD	sunshine duration [s]
t	time [s]
T	temperature [K]
u	augmented objective function [various units]
U	overall heat transfer coefficient [W/(m^2 K)]
V	volume [m^3]
$\dot{V}$	volume flow [m^3/s]
w	weight [-]

Greek Symbols:

α	empirical coefficient [-], regularization term [various]
Δt	time step [s]
ΔT	temperature difference [K]
η	efficiency [-]
θ	angle of incidence [°]
$\boldsymbol{\lambda}$	LAGRANGE multiplier vector [various]
ξ	exploration term [various]
ρ	density [kg/m^3], penalty term [various]

Abbreviations:

AI	artificial intelligence
AIC	AKAIKE information criterion
AL	augmented LAGRANGIAN
ANN	artificial neural networks
ARIMA	autoregressive integrated moving average
BO	Bayesian optimization
CNN	convolutional neural network
CV	cross-validation
DL	deep learning
GA	genetic algorithm
HGBR	histogram-based gradient boosting regressor
HP	heat pump
IPOPT	Interior Point OPTimizer
KNN	*k*-nearest neighbors
LSTM	long short-term memory
MAE	mean absolute error
ML	machine learning
MLR	multiple linear regression
MPR	multiple polynomial regression
PSO	particle swarm optimization
PV	photovoltaic
Rec	recursive
RFR	random forest regression
RHA	rolling horizon approach
STC	solar thermal collector
SVR	support vector regression
TES	thermal energy storage

References

1. Lund, H.; Werner, S.; Wiltshire, R.; Svendsen, S.; Thorsen, J.E.; Hvelplund, F.; Mathiesen, B.V. 4th Generation District Heating (4GDH): Integrating smart thermal grids into future sustainable energy systems. *Energy* **2014**, *68*, 1–11. [CrossRef]
2. Olatomiwa, L.; Mekhilef, S.; Ismail, M.; Moghavvemi, M. Energy management strategies in hybrid renewable energy systems: A review. *Renew. Sustain. Energy Rev.* **2016**, *62*, 821–835. [CrossRef]
3. Khalid, M. Smart grids and renewable energy systems: Perspectives and grid integration challenges. *Energy Strategy Rev.* **2024**, *51*, 1–26. [CrossRef]
4. Birge, J.R.; Louveaux, F. *Introduction to Stochastic Programming*; Springer Science & Business Media: New York, NY, USA, 2011.
5. Bertsimas, D.; Sim, M. The Price of Robustness. *Oper. Res.* **2004**, *52*, 35–53. [CrossRef]
6. Wu, H.; Xu, Z. Fuzzy Logic in Decision Support: Methods, Applications and Future Trends. *Int. J. Comput. Commun. Control* **2021**, *16*, 4044. [CrossRef]
7. Mavrotas, G.; Florios, K.; Vlachou, D. Energy planning of a hospital using Mathematical Programming and Monte Carlo simulation for dealing with uncertainty in the economic parameters. *Energy Convers. Manag.* **2010**, *51*, 722–731. [CrossRef]
8. Silvente, J.; Kopanos, G.M.; Pistikopoulos, E.N.; Espuña, A. A rolling horizon optimization framework for the simultaneous energy supply and demand planning in microgrids. *Appl. Energy* **2015**, *155*, 485–501. [CrossRef]

9. Bischi, A.; Taccari, L.; Martelli, E.; Amaldi, E.; Manzolini, G.; Silva, P.; Campanari, S.; Macchi, E. A rolling-horizon optimization algorithm for the long term operational scheduling of cogeneration systems. *Energy* **2019**, *184*, 73–90. [CrossRef]
10. Corinaldesi, C.; Schwabeneder, D.; Lettner, G.; Auer, H. A rolling horizon approach for real-time trading and portfolio optimization of end-user flexibilities. *Sustain. Energy Grids Netw.* **2020**, *24*, 100392. [CrossRef]
11. Cuisinier, É.; Lemaire, P.; Penz, B.; Ruby, A.; Bourasseau, C. New rolling horizon optimization approaches to balance short-term and long-term decisions: An application to energy planning. *Energy* **2022**, *245*, 122773. [CrossRef]
12. Erdinç, F.G. Rolling horizon optimization based real-time energy management of a residential neighborhood considering PV and ESS usage fairness. *Appl. Energy* **2023**, *344*, 121275. [CrossRef]
13. Kallabis, T. Rolling-Horizon Optimization As a Speed-Up Method—Assessment Using the Electricity System Model JMM. Technical Report, HEMF Working Paper No. 06/2020. 2020. Available online: https://ssrn.com/abstract=3684538 (accessed on 6 June 2025).
14. Bio Gassi, K.; Baysal, M. Analysis of a linear programming-based decision-making model for microgrid energy management systems with renewable sources. *Int. J. Energy Res.* **2022**, *46*, 7495–7518. [CrossRef]
15. Tawarmalani, M.; Sahinidis, N.V. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*; Nonconvex Optimization and Its Applications; Springer: New York, NY, USA, 2002; Volume 65. [CrossRef]
16. Li, S.; Yang, J.; Song, W.; Chen, A. A Real-Time Electricity Scheduling for Residential Home Energy Management. *IEEE Internet Things J.* **2019**, *6*, 2602–2611. [CrossRef]
17. Mquqwana, M.A.; Krishnamurthy, S. Particle Swarm Optimization for an Optimal Hybrid Renewable Energy Microgrid System under Uncertainty. *Energies* **2024**, *17*, 422. [CrossRef]
18. Kyriakidis, L.; Mendez, M.A.; Bähr, M. A hybrid algorithm based on Bayesian optimization and Interior Point OPTimizer for optimal operation of energy conversion systems. *Energy* **2024**, *312*, 133416. [CrossRef]
19. Kyriakidis, L.; Martin, B.; Mendez, M.A. Enhanced Hybrid Algorithm based on Bayesian Optimization and Interior Point OPTimizer for Constrained Optimization. *Optim. Eng.* **2025**, 1–52. [CrossRef]
20. Kansara, R.; Roldán Serrano, M.I. Coupled Design and Operation Optimization for Decarbonization of Industrial Energy Systems Using an Open-Source In-House Tool. *Eng* **2024**, *5*, 3033–3048. [CrossRef]
21. Solar Keymark Database. Annex to Solar Keymark Certificate. 2025. Available online: https://www.duurzaamloket.nl/DBF/PDF_Downloads/DS_305.pdf (accessed on 6 June 2025).
22. Canadian Solar. HiKu5 Mono Technical Information Sheet. 2022. Available online: https://cdn.enfsolar.com/z/pp/acc60dbda3e4ed35/5fe2bced1c3d5.pdf (accessed on 6 June 2025).
23. King, D.L.; Gonzalez, S.; Galbraith, G.M.; Boyson, W.E. Performance Model for Grid-Connected Photovoltaic Inverters. Technical Report SAND2007-5036, Sandia National Laboratories. 2007. Available online: https://energy.sandia.gov/wp-content/gallery/uploads/Performance-Model-for-Grid-Connected-Photovoltaic-Inverters.pdf (accessed on 6 June 2025).
24. DLR Solar Research. Greenius Manual. 2024. Available online: https://www.dlr.de/de/sf/forschung-und-transfer/forschungsdienstleistungen/simulation-und-wirtschaftlichkeitsbewertung/copy_of_greenius/greenius-support (accessed on 6 June 2025).
25. Jacobson, M.Z.; Jadhav, V. World estimates of PV optimal tilt angles and ratios of sunlight incident upon tilted and tracked PV panels relative to horizontal panels. *Sol. Energy* **2018**, *169*, 55–66. [CrossRef]
26. Cadau, N.; De Lorenzi, A.; Gambarotta, A.; Morini, M.; Rossi, M. Development and Analysis of a Multi-Node Dynamic Model for the Simulation of Stratified Thermal Energy Storage. *Energies* **2019**, *12*, 4275. [CrossRef]
27. Wagner, W.; Pruss, A. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *J. Phys. Chem. Ref. Data* **2002**, *31*, 387–535. [CrossRef]
28. Picheny, V.; Gramacy, R.B.; Wild, S.; Digabel, S.L. Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In Proceedings of the 30th International Conference on NIPS, Barcelona, Spain, 5–10 December 2016; pp. 1443–1451. Available online: https://papers.nips.cc/paper_files/paper/2016/hash/31839b036f63806cba3f47b93af8ccb5-Abstract.html (accessed on 6 June 2025).
29. Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M.L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. [CrossRef]
30. Deutscher Wetterdienst. Climate Data Center. 2025. Available online: https://opendata.dwd.de/climate_environment/CDC/ (accessed on 6 June 2025).
31. Gallo, R.; Castangia, M.; Macii, A.; Macii, E.; Patti, E.; Aliberti, A. Solar radiation forecasting with deep learning techniques integrating geostationary satellite images. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105493. [CrossRef]
32. Sehrawat, N.; Vashisht, S.; Singh, A. Solar irradiance forecasting models using machine learning techniques and digital twin: A case study with comparison. *Int. J. Intell. Netw.* **2023**, *4*, 90–102. [CrossRef]
33. Koller, M.; Feldmaier, J.; Diepold, K. A Comparison of Prediction Algorithms and Nexting for Short Term Weather Forecasts. *arXiv* **2019**, arXiv:1903.07512. [CrossRef]

34. Murtaugh, P.A. In defense of P values. *Ecology* **2014**, *95*, 611–617. [CrossRef] [PubMed]
35. Wan, X. Influence of feature scaling on convergence of gradient iterative algorithm. *J. Phys. Conf. Ser.* **2019**, *1213*, 032021. [CrossRef]
36. Tao, T.; Shi, P.; Wang, H.; Yuan, L.; Wang, S. Performance Evaluation of Linear and Nonlinear Models for Short-Term Forecasting of Tropical-Storm Winds. *Appl. Sci.* **2021**, *11*, 9441. [CrossRef]
37. Berrar, D. Cross-Validation. In *Encyclopedia of Bioinformatics and Computational Biology*; Ranganathan, S., Gribskov, M., Nakai, K., Schönbach, C., Eds.; Academic Press: Oxford, UK, 2019; pp. 542–545. [CrossRef]
38. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Louppe, G.; Prettenhofer, P.; Weiss, R.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
39. Mustafa Qamaruddin. Cross-Validation Strategies for Time Series Forecasting. 2019. Available online: https://medium.com/sci-net/cross-validation-strategies-for-time-series-forecasting-9e6cfab91f60 (accessed on 6 June 2025).
40. An, N.H.; Anh, D.T. Comparison of Strategies for Multi-step-Ahead Prediction of Time Series Using Neural Network. In Proceedings of the International Conference on Advanced Computing and Applications (ACOMP), Ho Chi Minh City, Vietnam, 23–25 November 2015; pp. 142–149. [CrossRef]
41. Rodriguez, H.; Medrano, M.; Rosales, L.M.; Peñuñuri, G.P.; Flores, J.J. Multi-step forecasting strategies for wind speed time series. In Proceedings of the IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 4–6 November 2020; Volume 4, pp. 1–6. [CrossRef]
42. SMARD. Marktdaten. 2025. Available online: https://www.smard.de/home/downloadcenter/download-marktdaten/?downloadAttributes=%7B%22selectedCategory%22:3,%22selectedSubCategory%22:false,%22selectedRegion%22:false,%22selectedFileType%22:false%7D (accessed on 6 June 2025).
43. Solgi, R.M. Geneticalgorithm 1.0.2. 2020. Available online: https://pypi.org/project/geneticalgorithm/ (accessed on 6 June 2025).
44. Panigrahi, B.K.; Shi, Y.; Lim, M.H. (Eds.) *Handbook of Swarm Intelligence: Concepts, Principles and Applications*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 8. [CrossRef]