**University of West Attica Heriot-Watt University**

School of Engineering and Physical Sciences MSc in Energy Systems

Dissertation

**Title: Multi-Objective Optimization of Photovoltaic Systems: Balancing Energy Production, Load Matching, and Economic Performance Through Advanced Modeling and Analysis**

**Author: Konstantinos Georgiou**

**Reg. Νumber: H00394220**

**Date:03/07/2025**

**Supervisor:Kavadias Kosmas**

# DECLARATION OF AUTHORSHIP

I, ..GEORGIOU KONSTANTINOS..............................................................

(name and registration number)

confirm that the report entitled ...................................................................

is part of my assessment for module **Β81ΕΖ**................................................................

I declare that the report is my own work. I have not copied other material verbatim except in explicit quotes, and I have identified the sources of the material clearly.

............................................................... Piraeus, ...................................................

(Signature) (Place and Date)

# ABSTRACT

Interest in maximizing photovoltaic (PV) system performance, both in terms of energy efficiency and economic viability, has increased due to the rising demand for sustainable energy solutions, with this thesis offering a thorough approach to PV system design and optimization through the integration of multi-objective optimization techniques, consumption forecasts, and physical energy production modeling, creating a comprehensive methodology, including seasonal analysis of performance and storage needs, system loss characterization, consumption-weighted mismatch minimization, and modeling of solar position and irradiance.

A multi-objective approach based on evolutionary algorithms was used for optimization, balancing demand matching and energy output, with the feasibility of the improved configurations being evaluated by an economic analysis that included cash flow and investment modeling. The results showed that the technical performance and financial returns of PV installations can be greatly improved by carefully optimizing tilt and azimuth angles, system sizing, and seasonal factors, with sensitivity analysis providing additional evidence that the results were resistant to changes in the parameters, providing a flexible and repeatable toolkit to academics, engineers, and stakeholders looking to optimize the advantages of PV deployments in practical operating environments.

Keywords: Photovoltaic Systems; Solar Energy Modeling; Multi-Objective Optimization; Genetic Algorithms; Energy Production Forecasting; Battery Sizing; Seasonal Analysis; Economic Viability; Renewable Energy Systems; DEAP Framework; PV Performance Optimization

# ACKNOWLEDGEMENTS

I would like to

# TABLE OF CONTENTS

[DECLARATION OF AUTHORSHIP 2](#_heading=h.jeo0lw1n6azd)

[ABSTRACT 3](#_heading=h.dd77vccm1od8)

[ACKNOWLEDGEMENTS 4](#_heading=h.evj3qo3mcllh)

[TABLE OF CONTENTS 5](#_heading=h.e27nu43usu8e)

[LIST OF FIGURES 7](#_heading=h.vuvhemzftqhp)

[LIST OF TABLES 8](#_heading=h.ulawtafjlyv9)

[GLOSSARY OF TERMS AND ACRONYMS 9](#_heading=h.t3prcqgnpidb)

[CHAPTER 1\_INTRODUCTION 10](#_heading=h.r6k6rrwnz8ja)

[1.1 Importance of Energy Production Simulation and Forecasting 11](#_heading=h.kjxhi8z9472l)

[1.2 Objective of this thesis 11](#_heading=h.xql9iucrebik)

[1.3 Structure of the Thesis 12](#_heading=h.u8jk8yddc2g2)

[CHAPTER 2\_Literature review 15](#_heading=h.k48t19a0nxep)

[2.1 Overview of Photovoltaic Technology 16](#_heading=h.z2322xp7m2w7)

[2.1.1 Evolution of PV Technology 16](#_heading=h.ezvp269eara)

[2.1.2 Current Trends in Solar Energy Research 17](#_heading=h.5lmz6r6p7prk)

[2.1.3 Key Challenges and Advancements 18](#_heading=h.uipnd8n67of9)

[2.2 Principles of Solar Energy Conversion 20](#_heading=h.uq5s4enoye)

[2.2.1 Photovoltaic Effect and Efficiency Factors 20](#_heading=h.pyqyu8dxyz1v)

[2.2.2 Solar Irradiance and Energy Harvesting 21](#_heading=h.46ci4kssg5gj)

[2.2.3 Conversion Losses and Performance Metrics 22](#_heading=h.8ujjwnvey4wp)

[2.3 Types of PV Systems and Their Components 24](#_heading=h.cj6x11z183kf)

[2.3.1 Grid-Connected PV Systems 24](#_heading=h.xyysz3k5l3l6)

[2.3.2 Off-Grid PV Systems 25](#_heading=h.34fop68g7dwb)

[2.3.3 Hybrid PV Systems 26](#_heading=h.u7lb8sw4g8co)

[2.3.4 Key Components of PV Systems 27](#_heading=h.x41pf07y8fsm)

[2.4 Energy Production Modeling in PV Systems 43](#_heading=h.ptg8173tczg4)

[2.4.1 Empirical Models (Data-Driven Models, Regression-Based Models) 43](#_heading=h.2cc82qo5mx0a)

[2.4.2 Physical Models (Meteorological and Irradiance-Based Models) 44](#_heading=h.53l1sk4o9l41)

[2.4.3 Hybrid Models (Combining Physical and Machine Learning Approaches) 46](#_heading=h.68wug7razpvk)

[2.4.4 Simulation Tools (pvlib, PVsyst, SAM, HOMER, MATLAB-Based Tools) 47](#_heading=h.mazuu8jts5)

[2.5 Optimization Techniques in Renewable Energy 49](#_heading=h.8pqcamhpuvbr)

[2.5.1 Evolutionary Algorithms (Genetic Algorithms, Particle Swarm Optimization, DEAP) 49](#_heading=h.qrwqjzv3sjd1)

[2.5.2 Mathematical Optimization Methods (Linear/Non-linear Programming, Convex Optimization) 50](#_heading=h.bwuhga9hys87)

[2.5.3 Machine Learning-Based Optimization (Neural Networks, Reinforcement Learning) 52](#_heading=h.yzi1pfdjyhre)

[2.5.4 Multi-Objective Optimization for PV Performance (Efficiency, Cost, Storage Integration) 53](#_heading=h.6pgwin3774xz)

[2.6 Forecasting Energy Consumption: Methods and Challenges 55](#_heading=h.fu20z78vihiy)

[2.6.1 Statistical Methods (ARIMA, Exponential Smoothing) 55](#_heading=h.xxuamvjucrb7)

[2.6.2 Machine Learning Models (ANNs, LSTMs, Hybrid Approaches) 56](#_heading=h.8xx4vu6bv7id)

[2.6.3 Data Requirements and Preprocessing (Time Series Data, Weather Dependencies) 58](#_heading=h.kxmt0jblwrzq)

[2.6.4 Challenges and Future Directions 60](#_heading=h.uk2lgq6d4hqt)

[CHAPTER 3\_Methodology 63](#_heading=h.houf8smtuizq)

[3.1 Core Research Questions 63](#_heading=h.agea8aj4a6a2)

[3.2 Methodological Framework 65](#_heading=h.tyfumcvyy9ps)

[3.3 Multi-Objective Optimization Approach 67](#_heading=h.rvf9ip8mcit1)

[3.3.1 Objective Functions 68](#_heading=h.h3hfuoywvshh)

[3.3.2 Genetic Algorithm Implementation 69](#_heading=h.ua6kbgega8ld)

[3.3.3 Parallelization Strategy 71](#_heading=h.w8srtp4r4m2m)

[3.4 Detailed Physical Modeling 72](#_heading=h.z48rsw5wv407)

[3.4.1 Solar Resource and Position Calculation 72](#_heading=h.q4xxis7v9i8z)

[3.4.2 Irradiance Modeling 73](#_heading=h.iaapup17su9b)

[3.5 Panel Performance Model 75](#_heading=h.5u0myztu3mw)

[3.6 System Losses Model 76](#_heading=h.weevj6s0588b)

[3.7 Consumption-Based Weighting System 76](#_heading=h.m1k102gmy3cc)

[3.7.1 Weighting Factor Calculation 76](#_heading=h.vnq3fvrjhlux)

[3.7.2 Weighted Mismatch Calculation 77](#_heading=h.9bl8dftt0meu)

[3.8.2 Balanced Solution Selection Algorithm 78](#_heading=h.q9ya2nn24b69)

[3.9 Seasonal Analysis Methodology 79](#_heading=h.amnnsrsc6639)

[3.9.1 Seasonal Categorization 79](#_heading=h.1ap7cmspxi88)

[3.9.2 Seasonal Metrics Calculation 79](#_heading=h.5x2iwmxykwcy)

[3.9.3 Seasonal Battery Requirements 80](#_heading=h.mzkd97r04yy2)

[3.10 Battery Sizing Methodology 80](#_heading=h.f21e9m74lgcb)

[3.10.1 Battery Simulation Parameters 80](#_heading=h.1058bzcsuio)

[3.10.2 Energy Flow Simulation Algorithm 81](#_heading=h.v3t4bjsb9v1e)

[3.10.3 Metrics for Optimization 81](#_heading=h.91yzny7pz8vs)

[3.10.4 Optimal Capacity Selection Criteria 82](#_heading=h.t96t1ywn3o7r)

[3.11 Economic and Efficiency Analysis 82](#_heading=h.usfg79q94ot6)

[3.11.1 Initial Investment Calculation 82](#_heading=h.j9dlagvtq3ts)

[3.11.2 Annual Cash Flow Modeling 82](#_heading=h.kdpipd5e1ncd)

[3.11.3 Financial Metrics 83](#_heading=h.1mmc59vd1c3u)

[3.11.4 Efficiency Metrics 83](#_heading=h.3mfp83kp7t2m)

[3.12 Integrated Results and Interpretation 84](#_heading=h.y3eorcy0knj)

[3.13 Energy Consumption Forecasting 85](#_heading=h.yr764yz4f8cp)

[CHAPTER 4\_ Application 87](#_heading=h.u2xqixlh6mnl)

[4.1 Code Organization & Structure 87](#_heading=h.u43pefd3pvh)

[4.2 Key Modules & Their Purpose 88](#_heading=h.mxm5gauvk43x)

[4.2.1 Imports and Dependencies 88](#_heading=h.z9y09epcgfm5)

[4.2.2 Configuration & Constants 89](#_heading=h.ywgems6gvdi4)

[4.3 Core Data Processing Functions 89](#_heading=h.3hz44era1fyu)

[4.3.1 Data Loading & Preprocessing 89](#_heading=h.z0sb6x3qcw3d)

[4.3.2 Solar Position Calculation 90](#_heading=h.wugnbbn22mdf)

[4.3.3 DNI Calculation 91](#_heading=h.vorgch1pwk7f)

[4.4. Solar Energy Modeling Functions 92](#_heading=h.hsh4s5vusk46)

[4.4.1 Total Irradiance Calculation 92](#_heading=h.6485xbpjm9o3)

[4.4.2 Energy Production Calculation 92](#_heading=h.mcztobhfzyrq)

[4.5 Weighting and Consumption Functions 95](#_heading=h.4hulv7hjoh9u)

[4.5.1 Weighting Factor Calculation 95](#_heading=h.zg96r81gg7nb)

[4.6 Multi-Objective Optimization Implementation 96](#_heading=h.vt630jev9j7a)

[4.6.1 Objective Function 96](#_heading=h.xng34lqf5c17)

[4.6.2 Optimization Context & Worker Management 97](#_heading=h.lxpe6muza5xu)

[4.6.3 DEAP Optimization Setup 98](#_heading=h.217vjueu1cr)

[4.6.4 Custom NSGA-II Evolution 101](#_heading=h.z0pltth3vgr)

[4.7 Seasonal Analysis Implementation 102](#_heading=h.kxymviqmj19t)

[4.8 Battery Sizing and Simulation 104](#_heading=h.4te2l17739us)

[4.9 Economic Analysis Implementation 108](#_heading=h.2xyvmpcx6fa)

[4.10 Main Execution Workflow 113](#_heading=h.qr0jy39ydu22)

[4.11 Forecasting Module Implementation 117](#_heading=h.i9rrpugjzu19)

[CHAPTER 5\_Analysis 120](#_heading=h.rswlfwjiio5k)

[5.1 Visualization of Key Results 120](#_heading=h.vj3g242tg1s8)

[5.1.1 Energy Forecasting 125](#_heading=h.2pzpjbohtlqh)

[5.2 Practical implications for real-world PV applications 148](#_heading=h.9hozh33thbe)

[5.3 Limitations of the study 149](#_heading=h.xuy2oj35a27w)

[5.4 Meaning of the Findings in Relation to Other Work 150](#_heading=h.cp0vbdvyj4ei)

[CHAPTER 6\_Conclusions 152](#_heading=h.8nkxt0uuqyv7)

[REFERENCES 154](#_heading=h.6rno9ebccwwi)

[APPENDICES 161](#_heading=h.e1bdbjr2g4oe)

Μορφοποίηση 161

# LIST OF FIGURES

# LIST OF TABLES

# GLOSSARY OF TERMS AND ACRONYMS

# CHAPTER 1 INTRODUCTION

Photovoltaic (PV) technology is rapidly expanding as a viable alternative to traditional fossil fuels due to the growing demand for sustainable energy solutions worldwide, with solar energy being essential to the shift to a low-carbon economy due to its abundance and renewable nature. But even with major advances in PV technology, issues with cost-effectiveness, efficiency, and grid integration still exist, with the performance and dependability of PV systems being impacted by the intrinsic fluctuation of solar radiation, determined by variables like location, weather, and seasonal variations, thus optimizing PV system performance has emerged as a essential study subject, aiming to optimize energy output and guarantee steady integration into power networks [1].

Improving panel efficiency, cutting energy losses, and boosting power forecasting accuracy are just a few of the many facets of PV system optimization, with practical application still facing obstacles such temperature fluctuations, shading effects, and deterioration over time, even though solar cell efficiency has grown due to technological advancements. Important elements that affect total system performance include choosing appropriate inverters, integrating energy storage technologies, and strategically positioning and orienting PV panels, making computational modeling and simulation essential in assessing these factors and creating plans to maximize energy output [2].

Furthermore, precise forecasting of energy generation and consumption is very important for maintaining supply and demand balance as solar energy becomes more integrated into power systems, with reliable predictive models being necessary for grid operators and energy regulators to effectively integrate PV-generated electricity and manage volatility. The nonlinear relationships between solar radiation and load demand are frequently difficult for traditional energy forecasting techniques to account for, with approaches based on artificial intelligence and machine learning allowing for more accurate forecasting models that can adjust to shifting circumstances, making the PV systems more resilient, economical, and efficient in today's energy infrastructures [3].

## 1.1 Importance of Energy Production Simulation and Forecasting

Optimizing the efficiency and dependability of photovoltaic (PV) systems requires precise energy production simulation and forecasting, with the production of solar energy being intrinsically erratic and affected by a number of variables, including the amount of sunlight, the climate, and seasonal variations, making it difficult to forecast how much electricity a photovoltaic system will generate at any given moment without accurate modeling, thus complicating system planning and grid integration. Researchers, engineers, and politicians can optimize system design and operation by using simulation tools to examine PV performance under various circumstances, aiding in making better decisions about panel installation, inverter selection, and storage integration by taking into account real-world factors like temperature, solar irradiation, and module efficiency [4].

It is equally important to forecast energy production, especially when integrating solar power into the electrical grid, as PV systems rely on changing external circumstances, making the power supply unpredictable, in contrast to traditional power plants that produce a steady output. Grid operators may prevent energy shortages, manage supply-demand mismatches, and lessen their dependence on backup fossil fuel-based power sources by using accurate forecasts, while aiding in infrastructure planning and investment decisions, short-term forecasting—which projects energy output within minutes to hours—is essential for real-time grid stability [5].

Forecasting energy output has greatly improved thanks to developments in artificial intelligence and machine learning, having transcended conventional statistical techniques. Large datasets can be processed, unveiling complex patterns, and making more precise predictions by using neural networks, recurrent models like LSTMs, and hybrid AI techniques. In addition to improving operational effectiveness, these approaches lower energy storage and grid stabilizing expenses, with accurate PV system forecasting and simulation being essential to the smooth integration of solar power into contemporary energy networks as the world's energy landscape moves toward renewables.

## 1.2 Objective of this thesis

The main goal of this research is to create a thorough methodology for improving the performance of photovoltaic (PV) systems using sophisticated forecasting, optimization, and simulation tools, aiming to improve the precision and dependability of PV system modeling in light of the fluctuating nature of solar energy production and the growing need for effective renewable energy solutions. The project also aims to address important issues in energy production prediction, system optimization, and consumption forecasting by combining computational tools and machine learning techniques, under the specific research goals:

1. Create a sophisticated framework for simulating PV systems that can precisely represent energy generation in a range of operational and environmental scenarios.
2. Use optimization strategies, such as evolutionary algorithms, to raise PV systems' dependability and efficiency.
3. Improve energy production forecasting by applying machine learning techniques to make very accurate short- and long-term predictions about solar power generation.
4. Analyze the precision, computational efficiency, and suitability for actual PV systems of various modeling and forecasting techniques to determine how effective they are.
5. Give researchers, politicians, and industry experts useful advice on how to best integrate PV systems into contemporary energy infrastructures.

## 1.3 Structure of the Thesis

The six chapters that make up this thesis each focus on a different facet of forecasting, optimization, and simulation for photovoltaic (PV) systems, with the backdrop and rationale for PV system optimization being given in Chapter 1: Introduction, also emphasizing the significance of precise energy production predictions and modeling, while also providing a clear direction for the study by outlining the main contributions and research aims.

A thorough analysis of the body of research on PV technology, modeling strategies, optimization tools, and forecasting approaches is provided in Chapter 2: Literature Review, giving an overview of photovoltaic technology at the outset, covering both historical and contemporary developments in solar energy. The Principles of Solar Energy Conversion are then covered, outlining how photovoltaic cells transform sunlight into electrical power, with PV systems being divided into grid-connected and off-grid configurations in the Types of PV Systems and Their Components section, which also describes key system components such trackers, batteries, and inverters.

Energy Production Modeling in PV Systems is next examined in the literature review, which summarizes numerous methods for estimating energy output under various operational and environmental circumstances, covering the topic of optimization techniques in renewable energy, investigating heuristic and AI-based approaches to enhancing PV efficiency. Lastly, Energy Consumption Forecasting: Techniques and Difficulties examines cutting-edge forecasting methods, highlighting the difficulties in estimating energy use in dynamic settings.

The methodological approach employed in this study is explained in Chapter 3: Methodology, with an emphasis on forecasting, optimization, and simulation techniques, describing the computer models that are used to examine PV performance in various scenarios. To improve system efficiency, the Optimization of PV System Performance section presents machine learning-based optimization methods and evolutionary algorithms, with the section on Energy use Forecasting finally describing the statistical and artificial intelligence-based models used to forecast trends in energy use.

Chapter 4: Application shows the experimental results and applies the suggested approaches to a real-world dataset, giving a description of the dataset and study area, together with information on the data sources, geographical features, and pertinent environmental elements, with the gear, software, and programming libraries used for model training and simulations being described in the Experimental Setup and Computational Tools section. Energy production simulations and optimization findings are analyzed and provided in findings from Simulations and Optimizations, while, Model Performance Analysis finally analyzes several predictive models and assesses forecasting accuracy.

The results are interpreted and their wider ramifications are discussed in Chapter 5: Analysis, with key findings from the experimental results highlighted in the Interpretation of Findings section, with a focus on forecasting accuracy and PV system efficiency. In Comparison with Existing Models and Methodologies the suggested methods, are compared to the most advanced techniques. How the research findings might be used to improve solar energy integration into power grids is covered in the section titled Practical Implications for Real-World PV Applications, with the study's limitations finally addressing limitations such modeling assumptions, computing restrictions, and data availability.

The research findings are compiled in Chapter 6: Conclusions, also offering possible avenues for further investigation, outlining the study's main contributions in the Summary of Key Findings, which also highlights the study's influence on PV system optimization. How the research increases our understanding of PV simulation, optimization, and forecasting is highlighted in the section on Contributions to the Field of Renewable Energy and Optimization. Potential Future Directions for Research, also including suggestions for methodological advancements, wider uses of the suggested models, and possible areas for additional research.

# CHAPTER 2 Literature review

The increasing demand for clean and renewable energy sources worldwide has fueled major improvements in the field of photovoltaic (PV) systems in recent decades, necessitating a thorough understanding of PV technology, energy production models, optimization strategies, and forecasting approaches for the efficient use of solar energy. This chapter offers a systematic evaluation of the body of literature already written on these important subjects, grouping pertinent studies into thematic groups in order to give an organized overview of the present state of study. An Overview of Photovoltaic Technology opens the chapter, detailing the development of PV systems from their infancy to their current state of development, drawing attention to current developments in solar energy research, such as cost savings, efficiency gains, and new technologies, while also covering the major obstacles that still influence the area, like grid integration problems and energy storage constraints.

The basic principles regulating PV systems, such as the photovoltaic effect, efficiency factors, and the effect of solar irradiance on energy harvesting, are then explained in the Principles of Solar Energy Conversion section, examining typical performance losses that impact energy output, highlighting the significance of precise modeling and optimization methods. Grid-connected, off-grid, and hybrid PV installations are categorized under the Types of PV Systems and Their Components section, examining the benefits, difficulties, and common uses of each category as well as important system elements with an emphasis on their function in energy conversion and storage, including solar panels, inverters, batteries, and charge controllers. Energy Production Modeling, which permits precise simulation and performance prediction under various environmental conditions, is a crucial component of PV research, examining and contrasting the advantages and disadvantages of physical, hybrid, and empirical modeling techniques, while also highlighting the capabilities of popular simulation tools for PV system analysis, including pvlib, PVsyst, SAM, HOMER, and MATLAB-based solutions.

Researchers have investigated a variety of optimization techniques in renewable energy to optimize the effectiveness and economics of PV systems, dividing optimization techniques into three categories: mathematical optimization techniques (like linear and non-linear programming), machine learning-based approaches (like reinforcement learning and artificial neural networks), and evolutionary algorithms (like genetic algorithms and particle swarm optimization), also covering multi-objective optimization techniques, with an emphasis on cost reduction, storage integration, and efficiency gains. Several approaches for estimating energy consumption in PV-powered systems are examined in the last section, Forecasting Energy Consumption: Methods and Challenges, comparing machine learning models like artificial neural networks (ANNs) and long short-term memory (LSTM) networks with statistical methods like exponential smoothing and ARIMA, also covering difficulties of forecasting, with special attention paid to data needs, preprocessing procedures, and the impact of outside variables like weather fluctuation.

## 2.1 Overview of Photovoltaic Technology

### 2.1.1 Evolution of PV Technology

With Alexandre Edmond Becquerel's 1839 discovery of the photovoltaic effect, photovoltaic (PV) technology began to take shape in the 19th century, with the first silicon-based solar cell created at Bell Laboratories in 1954, having an efficiency of about 6%, but practical uses did not appear until the middle of the 20th century, when PV technology began with this discovery, paving the way for further developments in material science and production techniques [6].

Solar energy research and development increased in the 1970s as a result of the oil crisis, raising interest in alternative energy sources, with governments and academic organizations funding PV technology, leading to a steady increase in cost and efficiency, while at the same time the first large-scale PV installations were made, mostly for off-grid power generation and remote applications like satellites [7].

Due to breakthroughs in silicon wafer manufacture and economies of scale, PV manufacturing made tremendous advancements in the 1990s and early 2000s, with thin-film solar cells gaining popularity because they were less expensive to produce than conventional crystalline silicon cells, allowing for wider use in both residential and commercial settings, while feed-in tariffs and government incentives in nations like Japan and Germany were essential in hastening the deployment of PV [8]. The efficiency and price of PV technology have advanced significantly during the last 20 years, with studies on new materials, including perovskite solar cells, showing promise for even greater efficiency while cutting production costs. Meanwhile, the number of practical uses has increased due to developments in floating solar farms, tandem cell architectures, and bifacial solar panels.

### 2.1.2 Current Trends in Solar Energy Research

Recent years have seen a considerable advancement in solar energy research, with an emphasis on increasing photovoltaic (PV) system efficiency, lowering costs, and broadening the systems' range of uses, and the creation of high-efficiency solar cells, with a focus on multi-junction and tandem cell structures, being one of the most active research topics [9]. In lab settings, these systems, which integrate various photovoltaic materials to collect a wider spectrum of sunlight, have shown efficiencies of over 40%. Perovskite solar cells have drawn a lot of interest among new materials because of its potential for low manufacturing costs and quick efficiency gains, with researchers trying to improve their scalability and stability to facilitate commercial adoption.

Bifacial solar panels, which can produce power from both direct sunlight and reflected light from nearby surfaces, represent another significant research trend, being a desirable alternative for utility-scale solar farms since it has demonstrated significant performance improvements in field installations [10]. Similar to this, floating solar farms, sometimes known as "floatovoltaics," have become a viable way to overcome land shortages, especially in areas with high population densities, with researchers wanting to increase energy production, decrease water evaporation, and mitigate temperature-related efficiency losses by putting PV systems atop bodies of water.

Another important area of research has been the development of energy storage integration, with the development of high-capacity battery storage options, like solid-state and lithium-ion batteries, becoming essential for managing intermittency as solar electricity continues to seep into the grid. In order to improve energy availability and dependability, researchers are also looking at hybrid PV systems, integrating solar energy with other renewable resources like wind or hydrogen generation.

Artificial intelligence (AI) and machine learning are becoming more and more important in maximizing solar energy output, in addition to hardware advancements, with AI-driven models being utilized for real-time system optimization, predictive maintenance, and performance predictions in order to increase productivity and reduce operating expenses [11]. By using these methods, PV systems can adjust to shifting weather patterns and identify problems before they cause significant performance problems.

Lastly, environmental impact and sustainability are now major factors in solar energy research, forcing scientists to look into ways to recycle solar panels so that elements like silicon, silver, and rare metals may be recovered and utilized again to solve the problem of PV waste management. In order to further lessen the environmental impact of solar energy generation, efforts are also being undertaken to produce PV materials that are non-toxic and devoid of lead, propelling breakthroughs that will make PV systems more economical, efficient, and environmentally sustainable.

### 2.1.3 Key Challenges and Advancements

Even with photovoltaic (PV) systems' quick development and advancements in technology, a number of obstacles remain in the way of their broad use and long-term viability, with commercial silicon-based solar panels normally having efficiency values between 18% and 22%, while despite tremendous advancements in multi-junction and tandem solar cell research, global adoption is still hampered by their high production costs and scalability problems [12]. Furthermore, environmental elements like shade, dust buildup, and temperature changes can impair PV systems' real-world performance, leading to the invention of self-cleaning coatings and adaptive tracking systems to optimize energy capture.

The intermittent nature of solar power generation poses a significant obstacle to its integration into the electrical grid, with the weather and the day-night cycle causing fluctuations in solar energy production, in contrast to fossil fuel-based power plants that consistently produce electricity. Developments in energy storage technologies have been a key area of attention in order to overcome this, with new technologies such as flow batteries and hydrogen storage, as well as lithium-ion and solid-state batteries, are being developed to store excess solar energy for later use, resulting to demand-response systems and smart grid technologies being used to improve the stability and adaptability of solar-powered grids [13].

Another issue is the high upfront costs of installing PV systems and related infrastructure, especially in poor nations with little access to funding and government subsidies, and even though solar panels have become much more affordable in the last ten years, costs associated with inverters, mounting systems, energy storage, and grid integration can still be high, with roll-to-roll printing for perovskite solar cells and better recycling techniques for silicon-based panels are two examples of innovations in manufacturing processes and materials that are intended to further lower costs and increase the affordability of solar energy.

Concern over the effects of PV panel production and disposal on the environment is developing from a sustainability standpoint, with carbon emissions and resource depletion being two environmental effects of the extraction of materials used in solar cells, such as silicon, silver, and rare earth elements. Additionally, panel waste management is becoming a major concern as the first generation of large-scale solar farms approaches the end of its operational life, with the creation of lead-free and non-toxic materials as well as improvements in solar panel recycling technologies being essential stages in making PV technology more environmentally friendly [14].

In terms of technology, recent developments have significantly enhanced PV systems' performance and versatility, with increased energy production having been made possible by the invention of bifacial solar panels, which capture sunlight from both sides of the panel. Meanwhile, floating solar farms have opened up new deployment options in coastal regions and water reservoirs where land is scarce, while at the same time, artificial intelligence (AI) and machine learning enabled real-time monitoring, defect detection, and predictive maintenance, transforming the solar sector and assisting in the optimization of energy output and the reduction of operating expenses. All things considered, even if PV technology still faces many obstacles, continuous research and development in materials, energy storage, AI-driven optimization, and recycling techniques is opening the door to more effective, affordable, and ecologically friendly solar energy options, required to achieve long-term sustainability in the energy sector and quickening the global switch to renewable energy.

## 2.2 Principles of Solar Energy Conversion

### 2.2.1 Photovoltaic Effect and Efficiency Factors

The basic physical mechanism that permits the conversion of solar energy into electrical power is known as the photovoltaic (PV) effect, and photons from sunlight must strike a semiconductor material, usually silicon, and transfer their energy to the electrons in order for electrons to escape their atomic bonds [15]. An internal electric field inside the solar cell separates the electron-hole pairs that are produced as a result, producing an electric current, making PV technology very appealing for producing sustainable energy as it converts sunlight directly into electrical power without the need for moving parts or air pollution.

Numerous factors affect a solar cell's efficiency, which is the ratio of electrical power produced to incident sunlight power, with the semiconductor material's bandgap energy being among the most crucial, affecting which solar wavelengths can be absorbed and transformed into electrical energy. With a bandgap of 1.1 eV, silicon—the most widely used material in commercial solar cells—can absorb a sizable amount of the sun spectrum but still loses energy as heat, with scientists investigating other materials like perovskites and multi-junction cells to increase absorption over a wider range of wavelengths.

Recombination losses, which happen when excited electrons recombine with holes prior to contributing to the electrical current, are another important element influencing efficiency, coming in three primary forms: radiative (where energy is liberated as light), Impurities and flaws in the material induce Shockley-Read-Hall, while intense sunlight frequently causes Auger recombination, with the secret to improving cell performance being minimizing these losses through passivation layers and increases in material purity [16].

Efficiency is also influenced by a solar cell's temperature coefficient, with higher temperatures generally impairing a cell's performance since more thermal energy results in more carrier recombination and a lower output voltage, leading to research being done on cooling techniques including passive heat dissipation and novel materials with reduced temperature sensitivity. Efficiency can also be greatly impacted by optical losses, being brought on by the cell surface reflecting sunlight and the semiconductor material's insufficient absorption, necessitating the use of textured surfaces and anti-reflective coatings to minimize this and optimize light absorption, and to further improve absorption, some sophisticated PV designs use light-trapping strategies.

Even though the efficiency of commercial solar cells are currently between 18% and 22%, research is still being done to push these boundaries, researching new technologies like plasmonic nanostructures, quantum dots, and tandem cells to improve energy conversion and get around efficiency barriers, making solar power a more attractive alternative to conventional energy sources as a result of PV technology's steady advancement toward improved performance.

### 2.2.2 Solar Irradiance and Energy Harvesting

The power received from the sun in the form of electromagnetic radiation per unit area is known as solar irradiance, affecting how well photovoltaic systems work due to the direct impact that fluctuations in sunshine have on energy output. Three primary components make up solar irradiance, which is commonly expressed in watts per square meter (W/m2): direct irradiance, which is sunlight that reaches the surface unhindered; diffuse irradiance, which is sunlight that is scattered by clouds and other atmospheric particles; and reflected irradiance, which is light that is reflected from nearby surfaces, with global horizontal irradiance (GHI), which is the sum of all elements, being essential for PV system performance modeling and optimization [17].

Numerous variables, such as location, time of day, season, and weather, affect how much solar energy is accessible for gathering, with solar potential being substantially larger in equatorial regions—where the sun is almost overhead for the most of the year—than in polar regions, when the sun is low in the sky [18]. Seasonal differences are especially important because winter months have a higher angle of incidence and fewer daylight hours, resulting in decreased irradiance, while the amount of sunlight that reaches PV panels is further influenced by cloud cover, air pollution, and atmospheric factors including humidity and particles, resulting in a highly fluctuating production of solar energy.

PV systems employ a variety of tracking and optimization strategies to optimize energy collection, with fixed-tilt systems being limited in their capacity to adapt to changing conditions, while being constructed with an ideal angle to gather the most sunlight throughout the year. The energy yield is greatly increased by single-axis and dual-axis tracking systems, which dynamically change the angle of solar panels to follow the direction of the sun, particularly in areas with strong direct irradiance, requiring however more upkeep and having additional complexity. Some PV applications use sun concentrators to improve energy harvesting in addition to tracking, potentially increasing power output while using less semiconductor material by using lenses or mirrors to direct sunlight onto high-efficiency solar cells, and although concentrated photovoltaic (CPV) systems need accurate tracking devices to stay aligned with the sun, they are especially successful in locations with large levels of direct sunlight.

Integrating energy storage and grid management technologies is an essential component of energy harvesting, with energy storage devices like batteries or pumped hydro storage assisting balance supply and demand because solar irradiation varies during the day and is absent at night, maximizing the integration of solar energy into the electrical grid in grid-connected systems, thus guaranteeing steady voltage and frequency levels.

### 2.2.3 Conversion Losses and Performance Metrics

In addition to the intrinsic characteristics of solar cells, the losses that take place during the energy conversion process also affect a photovoltaic system's efficiency, falling into three categories: thermal, electrical, and optical losses, with each of these types of losses lowering the overall performance of the system.

When incoming sunlight is not completely absorbed by the solar cell, optical losses take place, as semiconductor materials are naturally reflective, some light is reflected off the surface, with manufacturers utilizing surface texturing techniques to boost light absorption and add anti-reflective coatings to minimize this phenomenon [19]. Furthermore, certain photons might not be absorbed by the material, particularly if their energy is not equal to the semiconductor's bandgap, with light-trapping structures and multi-junction solar cells being examples of advanced technologies that try to minimize these losses.

Inefficiencies in the production, transportation, and collection of charge carriers are the cause of electrical losses, with recombination, in which excited electrons recombine with holes prior to contributing to the electrical current, being a significant source of loss, which can happen at the material's surface (surface recombination), in the material's bulk (bulk recombination), or at semiconductor structural flaws. In order to mitigate these problems, manufacturers employ passivation layers to lower surface recombination and enhance material quality to eliminate flaws, making material engineering and circuit design components of PV performance, since the efficiency of charge collection is further diminished by shunt resistance (induced by leakage currents) and series resistance (produced by metal contacts and interconnections).

The temperature sensitivity of solar cells results in thermal losses, with an increase in temperature causing a PV module's open-circuit voltage (Voc) to drop, lowering overall efficiency, and measuring this, is the temperature coefficient, which expresses how much power output decreases with increasing temperatures. As a result, several strategies are implemented including ventilation systems, passive heat dissipation, and improved materials with reduced thermal sensitivity are being investigated to reduce thermal losses.

PV systems' performance is assessed and compared using a number of important performance parameters, with efficiency (η), or the proportion of incident solar energy transformed into usable electrical energy, being one of the most basic, and calculated as follows:

where Pout​ is the electrical power output and Pin​ is the solar power incident on the module.

Another important parameter that assesses the quality of a solar cell is the fill factor (FF), which compares the actual maximum power production to the theoretical maximum power based on VocVocVoc and short-circuit current (Isc), and is computed as follows:

where the voltage and current at the maximum power point (MPP) are denoted by Vmp and Imp, with better cell performance and less internal losses being indicated by a greater fill factor.

Performance ratios (PR) are frequently employed in real-world photovoltaic installations to evaluate how well a system performs in real-world scenarios in comparison to its theoretical maximum output, and it is a useful metric of PV system quality, taking into consideration all losses, including shading, soiling, temperature effects, and system inefficiencies.

## 2.3 Types of PV Systems and Their Components

### 2.3.1 Grid-Connected PV Systems

The most popular kind of solar energy installations, especially in urban and industrial regions, are grid-connected photovoltaic (PV) systems, providing electricity to homes, companies, and industries while also using the grid for backup power when solar output isn't enough because of their direct connection to the utility grid. Grid-connected PV systems' main benefit is their capacity to offset power prices and transmit excess energy into the grid via feed-in tariffs (FiTs) and net metering [20].

An inverter, a metering system, and solar panels make up a standard grid-connected PV system, converting the direct current (DC) electricity produced by the solar panels into alternating current (AC) in order to meet the voltage and frequency requirements of the electrical grid. Grid-connected PV systems rely on the grid to balance supply and demand, therefore they don't need energy storage (batteries) like off-grid systems need, with more and more grid-tied systems integrating battery storage as a result of recent developments in battery technology, in order to improve energy independence and resilience.

The affordability of grid-connected photovoltaic systems is one of its main advantages, and compared to off-grid systems, installation and maintenance costs are lower because costly batteries are not required. Additionally, by selling extra electricity back to the grid, households and businesses with grid-connected photovoltaic systems can lower their overall energy costs, guaranteeing the efficient use of solar energy, whether for immediate consumption or for grid contribution during peak production hours.

Grid-connected PV systems have various drawbacks despite their benefits, with large-scale PV integration possibly having an impact on grid stability, especially in areas with high solar penetration. Weather-related sudden variations in solar power can affect system dependability, necessitating the use of sophisticated grid management strategies like demand response, energy storage integration, and smart grid solutions, and the majority of grid-connected PV systems shut down for safety concerns in the case of a power outage, unless having a backup battery system or a sophisticated islanding mechanism permitting continuous operation during blackouts.

### 2.3.2 Off-Grid PV Systems

Standalone solar power systems that run separately from the electrical grid are known as off-grid photovoltaic (PV) systems, offering energy independence, enabling homes, companies, and vital infrastructure—like hospitals and telecom towers—to operate independently of traditional power sources, being especially useful in disaster-prone locations where grid dependability is an issue and in rural electrification efforts.

An inverter, batteries, a charge controller, and solar panels make up a standard off-grid photovoltaic system, with the charge controller regulating the direct current (DC) electricity produced by the solar panels to avoid overcharging or deep draining of the battery bank [21]. Even when there is little solar radiation, like at night or on overcast days, the batteries' stored energy guarantees a steady power supply, and the inverter transforms the DC electricity that has been stored into alternating current (AC) to power household appliances and other electrical loads.

Energy management and storage is one of the main issues with off-grid photovoltaic systems, with their design carefully taking battery capacity and efficiency into account in order to consistently supply energy demands because they are unable to draw electricity from the grid. Early off-grid systems frequently employed conventional lead-acid batteries, but because of their greater energy density, longer lifespan, and enhanced efficiency, lithium-ion batteries are becoming more and more popular in contemporary systems, while at the same time, hybrid systems that include wind turbines or diesel generators can supply backup power during prolonged times of low solar generation [22].

The adoption of off-grid technologies is also heavily influenced by cost factors, as they do away with reliance on grid infrastructure and energy suppliers, although having substantial initial cost of batteries and backup systems. However, off-grid solar solutions are becoming more economical and effective due to falling solar panel and battery costs, as well as technology developments in energy management and smart microgrids, with off-grid solar solutions being also being promoted by governments and international organizations, especially in poor nations, through financial initiatives, subsidies, and policy incentives. Off-grid PV systems provide a scalable and sustainable way to provide energy access in underserved areas in spite of these obstacles, getting more dependable and economical with advancements in energy efficiency, smart monitoring, and battery technology, thus remaining essential to attaining global electrification and energy independence as the need for decentralized energy solutions increases.

### 2.3.3 Hybrid PV Systems

In order to provide a more dependable and adaptable energy solution, hybrid photovoltaic (PV) systems integrate solar energy with other energy sources including batteries, diesel generators, wind turbines, or even grid connectivity, being made to get over the drawbacks of both grid-connected and off-grid photovoltaic systems, guaranteeing a steady supply of electricity even in situations where the amount of solar energy generated is insufficient because of bad weather or operating at night, especially where energy dependability is essential, such as in remote locations, industrial settings, and microgrid configurations [23].

Solar panels, an inverter, a battery bank, and an additional power source like a wind turbine or diesel generator make up a conventional hybrid PV system, with the electricity produced by the solar panels during the day being either consumed immediately, stored in batteries, or supplied into the grid (if connected). To ensure a steady supply of electricity, the system can automatically transition to stored battery power or another energy source when solar generation is low, with a feature of advanced hybrid systems being the smart energy management systems, which optimize energy distribution among various power sources based on current availability and demand.

The increased energy efficiency and dependability of hybrid PV systems is one of their main advantages, lessening reliance on a single energy supplier by combining several power sources, guaranteeing constant access to electricity, making them perfect for uses like telecommunications infrastructure, rural electrification, and vital institutions like hospitals where power outages can be hazardous or expensive, while lowering operating expenses and their negative effects on the environment [24].

Hybrid PV systems have drawbacks and complications despite their benefits, with the integration of diverse energy sources necessitating sophisticated system design and control techniques to guarantee smooth functioning. Although declining battery prices and advancements in energy storage technologies are making hybrid systems more economically feasible, battery storage in particular raises the initial investment cost, while also requiring to be regularly maintained and observed, particularly when mechanical parts like wind turbines or diesel generators are included.

Hybrid PV systems are becoming more complex and popular as renewable energy technology develops, with future energy infrastructure anticipated to heavily rely on the integration of solar energy, battery storage, and intelligent energy management, especially in off-grid communities, disaster recovery initiatives, and sustainable microgrid projects, and hybrid PV systems contributing to closing the gap between sporadic renewable energy sources and dependable, round-the-clock power solutions with further innovation.

### 2.3.4 Key Components of PV Systems

For photovoltaic (PV) systems to effectively produce, control, store, and distribute power, a number of crucial components are required, with the main components—solar panels, inverters, batteries, and charge controllers—all play vital roles in guaranteeing the dependability and efficiency of the system. Designing an ideal PV system for off-grid, grid-connected, or hybrid applications requires a basic understanding of these elements.

**Solar Panels**

Since they are in charge of using the photovoltaic effect to turn sunlight into electricity, solar panels, sometimes referred to as photovoltaic (PV) modules, are the essential part of any PV system, with the numerous interconnected solar cells producing direct current (DC) electricity when exposed to sunlight. Environmental conditions, manufacturing methods, and material composition are some of the elements that affect solar panel performance, with the type of panel chosen having a significant impact on durability, cost, and energy efficiency, therefore it's critical to choose the right technology for each application.

Based on the makeup of its cells, solar panels are generally divided into three primary categories: thin-film, polycrystalline, and monocrystalline, appropriate for various use scenarios as each type has unique benefits and drawbacks.

Monocrystalline Panels are known for their great purity and homogeneity, monocrystalline panels, which are composed of a single, continuous silicon crystal structure, having a higher efficiency (usually between 18% and 22%) [25]. They last for more than 25 years and take up less room than other kinds, being also more costly because of the intricate production process, with their strong absorption of light giving them a distinctly black appearance.

The solar industry holds monocrystalline panels in high regard due to their remarkable efficiency and extended lifespan, requiring a single, continuous silicon crystal to be meticulously cultivated is used to make these panels. The end effect is a homogeneous and consistent structure that facilitates easier electron mobility inside the substance, making monocrystalline solar panels more efficient than many other varieties, such as polycrystalline and thin-film panels, thanks to their uniformity, which normally ranges between 18% and 22%, thus using less land to generate the same amount of energy as other technologies because they can turn a higher proportion of sunlight into electricity.

Compared to other varieties, the production process for monocrystalline solar panels is more complex and resource-intensive, with a single silicon crystal formed at the start of this process, and it is then cut into thin wafers, with the fabrication process requiring a high degree of precision to guarantee that every wafer maintains the continuous crystal structure that characterizes the monocrystalline type. Monocrystalline panels are typically more costly than their equivalents, such as polycrystalline panels, which are formed from silicon crystals that are less regular and continuous, because of this intricate production process, with the higher efficiency and space-saving features of monocrystalline panels typically outweighing the greater expense.

The longevity of monocrystalline panels is yet another important benefit, with many manufacturers offering guarantees that ensure a specific level of performance for up to 25 years, and these panels are known to last for more than 25 years. Monocrystalline panels are a desirable option for solar energy systems in homes, businesses, and industries because of its endurance, guaranteeing that the investment will continue to yield electricity for decades to come.

Another feature that sets monocrystalline panels apart from other varieties is their aesthetic attractiveness, as they are usually pitch black in color due to their great purity and the way they absorb light, and their increased efficiency can be attributed to the way light interacts with the crystalline structure, improving light absorption and reflection. Compared to other panel types, which can appear less consistent, its sleek and uniform appearance is frequently chosen for rooftop installations because it complements a variety of architectural styles and offers a more visually pleasant appearance.

Monocrystalline panels' improved performance, efficiency, and aesthetic appeal make them a popular choice, especially for those looking to maximize energy production in constrained settings, even though their higher cost may be a factor for some customers, being especially well-suited for situations where space is limited, like residential rooftops or smaller commercial structures, because of their higher energy return per square meter.

Polycrystalline Panels have a cheaper manufacturing cost but a poorer efficiency (about 15%–18%) since they are composed of many silicon pieces that have been fused together, resulting to their bluish tinge because of the way light interacts with the silicon grains, being nonetheless cost-effective even if they are less efficient than monocrystalline panels for large-scale solar projects where space is not an issue.

Multicrystalline solar panels, sometimes referred to as polycrystalline solar panels, are more affordable than monocrystalline ones, with polycrystalline panels being made in a very different way, with silicon being melted to create a block of crystals that are subsequently divided into individual wafers. Polycrystalline panels are composed of numerous tiny silicon grains that have been fused together, as opposed to monocrystalline panels, which are composed of a single, continuous crystal structure, leading to a decreased efficiency, usually between 15% and 18%, which lowers the panel's overall capacity to convert sunlight into electricity because the barriers between the silicon grains might obstruct the free passage of electrons.

Polycrystalline panels' primary benefit is their reduced manufacturing costs and compared to monocrystalline panels, they are easier to create and use fewer resources, being therefore typically less expensive to manufacture, making them a desirable choice for large-scale solar projects where the overall cost of installation is a crucial consideration. Even though a monocrystalline panel requires more room to generate the same amount of power due to its lower efficiency, in applications where cost-effectiveness is a top concern and land or roof space is plentiful, the cost savings may exceed this drawback.

The way that light interacts with the silicon grains in polycrystalline panels gives them their characteristic blue hue, and because of the numerous silicon crystals, the panel's surface is less uniform than that of monocrystalline panels, resulting in varied light scattering. The distinctive bluish color of polycrystalline panels is caused by this scattering phenomenon, and although the aesthetic attractiveness of polycrystalline panels may vary from person to person, when the panels are placed on rooftops or in big arrays, where functionality is prioritized above looks, the visual difference is frequently less obvious.

Although polycrystalline panels have a somewhat shorter lifespan than monocrystalline panels, they are renowned for their dependability and durability, frequently lasting 20 to 25 years, deteriorating more quickly than their monocrystalline counterparts over time although many manufacturers provide performance warranties. Polycrystalline panels can still yield a good return on investment even though their efficiency is a little lower, particularly in large-scale solar projects where land area is not a constraint, being frequently utilized in utility-scale or commercial solar projects, where installation space is less of a concern and cost per watt of energy is the main consideration.

All things considered, polycrystalline solar panels provide an appealing alternative for solar projects that need reduced upfront expenditures while still producing dependable and regular power generation since they strike a reasonable mix between price and performance, and even though they might not be as efficient as monocrystalline panels, many solar energy providers choose them because of their affordability and consistent performance in large-scale installations, particularly in markets where economic considerations influence consumer choice.

Panels with thin films, are made of amorphous silicon, cadmium telluride, or copper indium gallium selenide that are deposited onto a substrate, such as glass, plastic, or metal, to create thin-film panels, which differ from crystalline-based panels, having a lower efficiency (usually 10%–14%) and may need more installation space to produce the same amount of power as crystalline panels, despite being flexible, lightweight, and performing well in low light [26]. They are frequently utilized in industrial installations where weight restrictions are an issue, portable applications, and rooftops with solar integration.

Amorphous silicon (a-Si), cadmium telluride (CdTe), and copper indium gallium selenide (CIGS) are among the semiconductor materials used to create thin-film solar panels, a unique type of photovoltaic technology, with these substances being applied to a substrate, possibly made of metal, plastic, or glass, to produce lightweight, flexible, and thin panels. Thin-film panels are distinguished by their capacity to be produced in a considerably thinner form, usually only a few micrometers in thickness, in contrast to crystalline-based panels, which are made from solid silicon wafers, having a distinct advantage in situations where flexibility and weight are essential because of this.

Thin-film panels' lower efficiency, which typically ranges between 10% and 14%, is one of its main characteristics, indicating that in order to generate the same amount of power as their crystalline equivalents, such as monocrystalline and polycrystalline panels, they need a larger surface area. When compared to crystalline silicon, the materials' less effective light absorption and conversion capabilities are the main cause of the decreased efficiency, being however prized for their cost and adaptability despite their poorer energy conversion performance.

Thin-film solar panels have clear benefits in some situations, even if their lower efficiency necessitates greater installation space to produce the same amount of electricity as crystalline panels, with their flexibility and light weight for example making them perfect for uses like industrial installations, rooftop solar integration, and portable solar devices where the weight of conventional panels would be too much, being especially helpful in circumstances where conventional crystalline panels would not be appropriate since they can be used on a range of surfaces, including curved or unusual shapes.

Thin-film panels are more flexible and lightweight than their crystalline counterparts, and they also function better in low light, being a good choice in areas where there is a lot of cloud cover or less direct sunshine, where other kinds of panels would not work as well. Additionally, thin-film panels are less susceptible to high temperatures, which can impair crystalline silicon panels' performance, and because of its thermal stability, thin-film technology is especially well-suited for hot climes where heat-induced losses could cause other panels to perform noticeably worse.

In general, the production process for thin-film panels is less complicated and costly than that of crystalline panels, with the reduced total cost of thin-film technology can be attributed to the fact that the materials utilized are frequently more affordable and readily available than the high-purity silicon needed for crystalline panels. Because of this, thin-film panels are a desirable choice for large-scale installations and projects where initial costs are a key factor, with their usage in residential applications where space is limited, nevertheless, because of their poorer efficiency, necessitating more space and materials despite the cost advantages.

Thin-film panels' affordability, adaptability, and capacity to function in less-than-ideal lighting circumstances make them popular in commercial applications such as solar farms, building-integrated photovoltaics (BIPV), and portable solar systems, being also a popular option for solar facades or roofing in urban settings because of their ability to blend in perfectly with architectural features.

Numerous operational and environmental factors affecting the efficacy and efficiency of solar panels:

* Solar Irradiance

One of the key elements affecting photovoltaic (PV) systems' effectiveness and performance is solar irradiation, with watts per square meter (W/m2) being a common unit of measurement for the quantity of solar energy received per unit area at any particular moment, having a direct effect on the PV modules' ability to generate energy since the intensity of the irradiance that solar panels receive determines how well they convert sunlight into electricity. The design, installation, and operation of solar energy systems depend heavily on solar irradiance, which is essentially the amount of sunshine that reaches the photovoltaic cells, enabling them to produce more power.

Geographical location, time of day, and seasonal variations all have a significant impact on how much sunshine solar panels receive, with higher levels of solar irradiance typically found in regions nearer the equator, which receive more direct sunlight all year round, often referred to as high-sunlight areas—being perfect for solar panel installations because they can optimize energy output, these areas. The potential energy output of PV systems may be constrained by the lower levels of solar irradiation found in locations nearer the poles, in regions with regular cloud cover, or in places with high levels of air pollution.

The incidence angle, which is the angle at which sunlight reaches the solar panel's surface, has a big impact on how much energy the panel absorbs, with solar panels being able to capture more sunlight during midday, when the sun is directly overhead and the irradiance is at its highest. Nevertheless, the angle of sunlight varies during the day as the sun moves across the sky, lowering the amount of irradiance that reaches the panels, leading to why large-scale installations occasionally employ solar tracking devices, modifying panel orientation to follow the sun, to boost energy capture and boost total efficiency. Solar irradiation is also impacted by seasonal changes, and because of longer days and a more direct angle of incidence, areas at higher latitudes usually receive more sunlight throughout the summer. In contrast, there is less solar irradiation during the winter months since the days are shorter and the sun's angle is shallower, and especially in the northern and southern latitudes, where solar panel energy production can vary significantly based on the season, this difference being noteworthy.

The amount of energy produced by solar panels can also be decreased by cloud cover, weather patterns, and atmospheric conditions, with clouds partially making up for the loss of direct sunshine by letting diffuse sunlight through even though they may block direct sunlight. However, the amount of solar energy that is accessible can be greatly reduced on cloudy days or during periods of intense rain, which lowers panel efficiency. Furthermore, sunlight can be scattered by dust, pollution, and other airborne particles, which lowers the amount of solar irradiance that reaches the panels and compromises their functionality, being especially noticeable since dust buildup on panel surfaces can impair performance even more in arid areas with high dust concentrations, such deserts.

Solar irradiation directly correlates with PV system energy production in terms of panel efficiency, with the optimal performance of solar panels being achieved when they are exposed to a steady and high degree of solar radiation. Nevertheless, the power output of the panels may change as irradiance levels change as a result of environmental conditions, with a PV system situated in an area with a lot of clouds, for example, producing less energy overall than one situated in an area with clear sky and lots of sunshine. As a result, making sure that solar panels are placed in locations that receive the most sunshine for the maximum potential for energy generation requires careful system design and site selection. Strategies for optimizing energy output can also be used to lessen the impact of varying sun irradiation, with batteries and other energy storage devices, for instance, storing extra energy produced during high-irradiance periods and release it during low-irradiance ones, guaranteeing a constant power output even when sunshine isn't at its strongest and helps to stabilize the energy supply. Furthermore, by tracking irradiance levels and modifying solar panel operation accordingly, performance monitoring systems can maximize efficiency and reduce losses brought on by environmental fluctuations.

* Temperature:

One crucial operational element that has a big impact on photovoltaic (PV) systems' effectiveness and performance is temperature, and even while solar panels need sunlight to produce power, it's a popular misperception that warmer temperatures necessarily translate into more efficient energy production. In actuality, solar panels' performance—especially that of the semiconductor elements they contain—can be adversely impacted by extremely high temperatures, based on the fact that electrical resistance increases with temperature, lowering their total power output.

Manufacturers use this standard testing setting to grade panel performance because photovoltaic cells work best at temperate temperatures, usually about 25°C (77°F), with the efficiency of solar cells starting to decrease as the surrounding temperature rises above this threshold. Most solar panels lose about 0.3% to 0.5% of their performance for each degree Celsius that the temperature rises above the ideal range, implying that a panel's output may be noticeably lower than its quoted capacity in extremely hot conditions, particularly in direct sunlight with no circulation.

The temperature sensitivity of semiconductor materials, especially silicon, which is utilized in the majority of solar panels, is the cause of this occurrence, as the silicon's band gap narrows with increasing temperature, lowering the voltage the panel can produce. Temperature may cause a modest rise in current, but the voltage loss is greater than the current gain, resulting in a net power drop, while in hotter climes or during the hottest summer months, heat acts as a sort of performance drag on the energy conversion process.

PV system designers use a number of heat management techniques to overcome this problem. Installing panels with sufficient ventilation space underneath them to promote circulation and aid in heat dissipation is a popular technique, with racking systems for instance raising the panels above the roof surface during rooftop installations so that air can flow and overheat is avoided. Furthermore, certain high-efficiency panels are constructed with coatings and materials that are better able to withstand heat, allowing them to continue operating even in extremely hot environments.

The location and orientation of the panels can also affect the system's heat load, with temperature effects being controlled, for instance, by positioning panels at angles that minimize extended sun exposure or prevent heat buildup at the warmest times of the day. The use of light-colored or reflective materials surrounding the panels can also aid in lowering ambient heating in large-scale solar farms, and when installed correctly, technologies like bifacial panels, which can benefit from reflected light from the ground, potentially provide better performance with less thermal depreciation.

It's also critical to keep in mind that different panel types have different temperature effects, with thin-film panels for example, often performing better in high-temperature situations because of their lower temperature coefficients, are generally more heat-sensitive than monocrystalline and polycrystalline panels. Although thin-film technologies may have a lower overall efficiency under normal conditions, this makes them more appropriate for some desert or tropical sites.

Lastly, sophisticated monitoring systems that monitor panel temperatures and adjust output appropriately can preserve or enhance system performance, and in order to optimize efficiency, these systems can recommend real-time modifications or warn operators of overheating problems. Temperature-aware management, when combined with energy storage systems, can further improve system resilience and dependability.

* Shading:

One of the most important and frequently disregarded elements that can have a big influence on photovoltaic (PV) system performance is shading, with power production disproportionately reduced by even slight shade, such as that produced by utility poles, tree branches, chimneys, or accumulated dirt and debris, mainly because solar panels are made up of many interconnected photovoltaic cells, and even a small amount of shadowing on one of the cells can cause the electrical current to be disrupted throughout the entire string. Solar panels are connected in series in conventional PV systems, particularly those that use string inverters, indicating that the panel that generates the least quantity of electricity limits the string's overall electrical output. Consequently, the current passing through all of the panels in that string can be decreased if only one panel is partially shaded, and in the system design phase, it is important to reduce shadowing wherever feasible as it might result in significant efficiency losses.

The time of day, season, and location all affect how severe the shading effects are. For example, when the sun is lower in the sky in the early morning or late afternoon, shade is more likely to happen, while shading can become a constant problem that varies throughout the year in metropolitan settings or places with diverse topography. During the planning phase of a solar installation, a thorough site analysis utilizing solar pathfinders or software tools is crucial to reducing these problems. Modern solutions have been created to mitigate the negative effects of shade, with the usage of microinverters, which are mounted on separate panels as opposed to a single centralized inverter, being one such breakthrough. Because of this, each panel may function independently, therefore the output of the others is unaffected if one is shaded. In installations where partial shadowing is inevitable, like on residential rooftops with chimneys or close to trees, microinverters work very well. Using bypass diodes in the design of the solar panel is another practical alternative, keeping the shaded area from bottlenecking the entire string or panel by giving the current a different path around underperforming or darkened cells. Bypass diodes are frequently included in the majority of contemporary crystalline silicon panels and can greatly lessen the effect of shading. In order to minimize shading losses, strategic system design is just as vital as hardware solutions, entailing maximizing the panels' tilt and orientation, arranging them in the right spacing, and avoiding locations that are likely to be shaded all day or all year, with energy yield and system reliability potentially being further improved for large-scale installations by employing shade analysis tools for site selection and layout development. It is also important to remember that, particularly in arid or dusty settings, soiling and dirt buildup can serve as a type of diffuse shade, and maintaining system performance requires routine maintenance, such as panel cleaning routines, thus automated cleaning systems or anti-soiling coatings can be used in some situations.

* Degradation Over Time:

Over the course of their operating lifetime, solar panels, like all technologies, undergo wear and tear, and when assessing the long-term efficiency and return on investment of a photovoltaic (PV) system, deterioration over time—a phrase used to describe this natural drop in performance—is a crucial consideration, and although it can vary depending on the quality of materials, manufacturing techniques, and climatic conditions, solar panels typically deteriorate at a rate of 0.5% annually. Assuming this yearly degradation rate, a panel that was originally rated at 100% of its capacity will function at about 90% of that capacity after 20 years due to the degradation process, and although it might not seem like much, over decades it can have a big influence, especially for large installations where continuous energy production is essential to their financial sustainability. Because premium materials and cutting-edge production processes are used, high-quality panels often deteriorate more slowly, with monocrystalline panels for instance usually showing lower rates of deterioration, compared to polycrystalline or thin-film panels, as monocrystalline silicon is more resistant to long-term environmental stress because it contains fewer impurities and structural irregularities. To further protect the solar cells from moisture, ultraviolet (UV) light, and mechanical stress, all of which can hasten breakdown if ignored, protective coatings and encapsulants are also applied to panels.

Potential-induced degradation (PID), which can happen when there is a leakage of current from the photovoltaic cell to other parts of the panel due to high voltage stress, temperature fluctuations, humidity, mechanical stresses from wind and snow loads, and UV exposure are the main causes of degradation, leading to reduced performance. Manufacturers carry out extensive testing under accelerated aging circumstances to forecast the performance of panels over time in order to combat these problems, leading to the majority of trustworthy manufacturers providing performance warranties that, after 25 years of operation, guarantee a specific level of production, usually 80% or more of the initial capacity, giving system owners a certain amount of operational and financial security for the anticipated life of the panel. Degradation management also involves monitoring systems, with any irregularities or sudden declines in performance being promptly detected by tracking energy production continually, enabling prompt maintenance or panel replacement, potentially reducing losses over time and contributing to extending the PV system's useful service life.

* Angle and Orientation:

The amount of sunlight that solar panels receive is mostly determined by their orientation and angle, which has a direct impact on how efficiently and effectively they produce energy overall, requiring photovoltaic panels to be exposed to as much direct sunlight as possible since they convert sunlight into electrical energy to operate at their best. During system installation, the tilt angle (vertical inclination) and azimuth angle (horizontal orientation with respect to true north) of the panel must be carefully taken into account. Panels should be oriented to face true north in the southern hemisphere and true south in the northern hemisphere for permanent installations, thus receiving as much sunshine as possible throughout the day, especially during the noon hours when solar irradiance is at its highest. In order to balance summer and winter solar exposure, the tilt angle should also roughly match the installation site's latitude.

Seasonal sun trajectories, however, affect the ideal angles, and since the sun is lower in the sky during the winter, steeper tilt angles are necessary to properly capture sunlight. On the other hand, since the sun is higher in the summer, a flatter tilt might be more beneficial, with designers frequently choosing a compromise angle in fixed-mount systems in order to optimize the annual energy yield, modified appropriately for applications where optimizing output during a particular season is crucial, like as off-grid systems utilized in winter cabins. Tracking or adjustable mounting mechanisms can be employed to further enhance performance and handle seasonal volatility, and while automated single or dual-axis trackers continuously adjust the panel's position to follow the sun's movement throughout the day and year, manual tilt adjustment enables users to change the angle a few times a year, more frequently used in utility-scale installations than in residential ones due to their complexity, expense, and maintenance needs, raising energy output by 20% or more. Reduced sun exposure from improper tilt and orientation might result in decreased efficiency and lost potential with panels facing east or west for instance only getting the best sunlight for a portion of the day, and too shallow or steep an angle might result in dirt buildup or seasonal underperformance, both of which lower production. Furthermore, in residential systems, the installation angle is frequently limited by the pitch and orientation of the roof, requiring design trade-offs to optimize performance within physical limitations.

The durability and efficiency of solar panels have significantly increased as a result of ongoing research and development. Growing in popularity are bifacial solar panels, which use reflected sunlight to produce power from both sides, with perovskite solar cells having also demonstrated a lot of potential in terms of attaining high efficiency at reduced production costs, while at the same time, innovations like tandem solar cells, nanostructured materials, and anti-reflective coatings are increasing the efficiency and accessibility of solar energy by pushing the limits of PV technology.

**Inverters**

By transforming the direct current (DC) electricity produced by solar panels into alternating current (AC), the common kind of electricity used in residences, workplaces, and power grids, inverters are necessary components of photovoltaic (PV) systems, as PV modules' solar energy would not be compatible with the grid or traditional electrical appliances without inverters, with its functionality, dependability, and efficiency having a big impact on how well a PV system performs overall.

String inverters, microinverters, and hybrid inverters are the three primary inverter types that are frequently utilized in PV installations, with each type having unique qualities, benefits, and uses depending on the needs for energy management and system design [27]. The most popular inverters for grid-connected photovoltaic systems are string inverters, having a string of solar panels connected in series that feeds into a single centralized inverter, and are renowned for being reasonably priced and easy to install. The fact that the weakest panel determines the string's overall performance is a significant disadvantage, though, resulting in the effectiveness of the entire string declining if one panel is shaded, accumulates dirt, or degrades, resulting to string inverters being less suitable for installations with various panel orientations or levels of sunshine exposure.

On the other hand, microinverters are mounted on every single solar panel, enabling autonomous power optimization, guaranteeing that every panel performs to its full capacity, in contrast to string inverters, where shading or performance problems in one panel impact the entire system. Higher energy yields, better monitoring, and more system design flexibility are the outcomes of this, with numerous units required, however, leading to microinverters being typically more expensive than string inverters, with their greater initial cost being frequently justified when irregular roof patterns or shading would otherwise impair system performance.

For hybrid PV systems that combine solar panels and battery storage, hybrid inverters are made specifically for such systems, being perfect for self-consumption and energy independence applications since they can control solar energy generation, battery charging and discharging, and grid interface, and although hybrid inverters have more functionality than traditional string inverters, they are typically more complicated and expensive, and they require compatible battery systems.

**Batteries**

Because they allow energy to be stored for later use, batteries are essential to photovoltaic (PV) systems, storing extra solar energy produced during the day, enabling a steady power supply in situations where sunshine isn't available, like at night or in overcast weather in off-grid and hybrid photovoltaic systems, thus lessening the reliance on the electrical grid by ensuring energy self-sufficiency and reliability. Batteries also aid in energy efficiency in grid-connected systems by storing solar energy when output surpasses demand and releasing it during periods of high grid electricity rates.

Lead-acid and lithium-ion batteries are the two primary battery types that are frequently seen in PV systems, with each type being appropriate for a variety of uses Because of their unique qualities, benefits, and drawbacks [28]. Because of their cost and track record of dependability, lead-acid batteries have long been a popular option for solar energy storage, being frequently utilized in off-grid photovoltaic systems where economy is a top concern. Lead-acid batteries do have some disadvantages, though, such as a comparatively short lifespan, a lower energy density, and the requirement for frequent maintenance, lifespan that can be even more shortened by recurrent full depletion since they are susceptible to deep discharges, while also requiring to be handled carefully and with adequate ventilation because of the possibility of gas emissions and sulfuric acid leaks during cycles of charging and discharging.

In contrast, lithium-ion batteries have become the go-to energy storage option for contemporary photovoltaic systems, having a higher energy density, allowing them to store more energy in a lighter and more compact package compared to lead-acid batteries. A much longer lifespan is another benefit of lithium-ion technology, as it can withstand more charge-discharge cycles before degrading, being ideal for both residential and commercial solar applications also supporting improved efficiency and faster charging. Although they are more expensive initially than lead-acid batteries, they wind up being a more economical option in the long run due to their longer lifespan and lower maintenance needs.

In PV systems, a number of factors affect battery performance and overall efficacy, with deeper discharges potentially reducing battery lifespan, especially in lead-acid systems. Depth of discharge (DoD) is the proportion of a battery's total capacity that is utilized before recharging, with a battery's charge cycles indicating how many times it may be charged and drained before its capacity noticeably drops; lithium-ion batteries usually have more cycles than their lead-acid counterparts. Additionally, temperature is important since extremes of heat or cold can affect the longevity and efficiency of batteries, runing safely and optimally when the system is properly integrated with charge controllers and inverters.

**Charge Controllers**

For photovoltaic (PV) systems with battery storage, charge controllers are essential parts, controlling the electricity flow between solar panels and batteries, keeping them charged effectively and avoiding severe discharge or overcharging. As deep discharge can drastically shorten battery life, while overcharging can result in excessive heat and gas buildup, which may cause battery failure, charge controllers safeguard battery health and raise a PV system's overall dependability and efficiency by controlling the charging process.

Maximum Power Point Tracking (MPPT) controllers and Pulse Width Modulation (PWM) controllers are the two primary charge controller types utilized in solar energy systems, with each type being appropriate for varying system sizes and energy needs and has unique qualities [29]. The easier and more affordable choice is a pulse width modulation (PWM) controller, controlling battery charging by progressively lowering the current as the battery approaches full charge in order to avoid overcharging and maintain a constant charge level. Smaller PV systems with lesser power requirements, like off-grid cabins or standalone solar lighting, are the ideal fit for PWM controllers, posing a big disadvantage however as they don't modify the voltage to optimize power extraction from the solar panels, they are less effective than MPPT controllers, being less suitable for larger or more energy-intensive installations.

Conversely, Maximum Power Point Tracking (MPPT) controllers are more sophisticated and effective, dynamically modifying the operating settings to optimize energy harvesting while continually monitoring the voltage and current output of solar panels in contrast to PWM controllers, and can improve efficiency by up to 30% by optimizing the PV array's power point, especially in systems with higher voltage solar arrays. For bigger PV projects where optimizing energy conversion is essential, MPPT controllers are therefore the recommended option, being however more costly than PWM controllers, but for medium-to-large PV systems, their improved performance and efficiency make the difference.

## 2.4 Energy Production Modeling in PV Systems

### 2.4.1 Empirical Models (Data-Driven Models, Regression-Based Models)

Empirical models, sometimes referred to as data-driven models, estimate the energy production of photovoltaic (PV) systems using statistical methods and historical data, using observational data to determine the correlations between input factors (such as irradiance, temperature, and system parameters) and energy output, contrasting to physical models that rely on basic equations regulating solar radiation and system behavior, being especially helpful when a quick, computationally efficient estimate is needed or when comprehensive system details are not available.

Regression-based models, which create mathematical connections between input and output variables, are among the most popular kinds of empirical models, being frequently employed for straightforward energy output forecasts assuming a direct proportionality between variables like solar irradiance and power generation [30]. However, non-linear elements like shading and temperature impacts affect PV system performance, limiting the accuracy of simple linear models, with non-linear regression models and polynomial regression techniques being used to overcome these constraints and capture more intricate relationships between system characteristics.

Because machine learning-based empirical models can handle big datasets and simulate complex relationships, they have become more popular than classic regression methods, with PV energy output being frequently predicted using random forests, support vector machines (SVMs), and artificial neural networks (ANNs) using past weather and system performance data [31]. Compared to traditional regression methods, these models can adjust to changes in the environment and increase forecast accuracy, being better suited for large-scale PV forecasting applications, though, as they demand a lot of training data and processing power.

The main benefit of empirical models is their capacity to generate predictions quickly and with a reasonable degree of accuracy without necessitating in-depth understanding of system physics, being especially helpful for performance evaluation and short-term projection in PV facilities that are currently in operation. However, the caliber and volume of available data have a significant impact on their correctness, with empirical models potentially having trouble generalizing when system conditions differ greatly from the training dataset. Empirical and physical models are frequently merged in hybrid modeling approaches to increase forecast accuracy, utilizing the advantages of both approaches, improving the robustness of PV energy production simulations by combining data-driven methods with physical limitations, making them useful tools for optimizing solar power generation in a variety of applications.

### 2.4.2 Physical Models (Meteorological and Irradiance-Based Models)

The basic concepts of solar radiation, weather patterns, and the physical properties of PV systems are the foundation of physical models used to simulate photovoltaic (PV) energy production, with physical models employing deterministic equations to simulate how environmental elements, such as temperature, air conditions, and solar irradiation, affecting PV system performance, in contrast to empirical models, which rely on historical data to create statistical connections, being frequently used for system design, performance assessment, and long-term energy yield estimation since they provide a more flexible and broader approach than strictly data-driven techniques.

The solar irradiance model, which calculates how much solar energy reaches the PV panels, is an essential part of physical models, accounting for elements including cloud cover, shading effects, atmospheric attenuation, and the sun's position [32]. The Liu and Jordan model, which is widely used to predict solar radiation on tilted surfaces, and the Perez model, which estimates diffuse radiation under various sky conditions, are examples of commonly used irradiance models, assisting in determining the available solar energy input, thus accurately estimating the operation of PV systems.

The meteorological modeling of environmental influences, especially the effect of temperature on PV module efficiency, is a significant component of physical modeling, with PV cells losing efficiency as their temperature rises because of the inherent characteristics of semiconductor materials [33]. Power output projections based on wind speed and ambient temperature are adjusted using temperature correction models, such as those based on the temperature coefficient of power or NOCT (Nominal Operating Cell Temperature) models, potentially also including site-specific weather information, such as humidity, air mass, and wind influences To improve PV performance simulations.

Physical models, such as I-V curve modeling, take into account the electrical properties of PV modules, while also illustrating the relationship between current and voltage under various temperature and irradiance circumstances. The electrical behavior of PV cells is frequently represented by single-diode and double-diode models, which include factors like series and shunt resistances to replicate actual losses from internal resistance and recombination effects, being crucial for designing high-efficiency PV installations as they offer comprehensive insights into system performance.

The flexibility of physical models to mimic PV performance across a range of operating and environmental situations without the need for substantial historical data is one of their main advantages, being especially helpful for system sizing, performance analysis, and long-term energy yield estimation in new PV systems. However, the availability of accurate system parameters and high-quality meteorological data determines how accurate they are, with physical models frequently combined with machine learning methods or real-time monitoring data to increase reliability, resulting in hybrid models that combine the advantages of physics-based and empirical methods.

### 2.4.3 Hybrid Models (Combining Physical and Machine Learning Approaches)

Hybrid models enhance the precision and versatility of photovoltaic energy production simulations by combining the advantages of both physical and machine learning methodologies, and while empirical models use statistical relationships from historical data and physical models use deterministic equations based on system parameters and meteorological data, hybrid models combine these approaches to produce a more reliable and broadly applicable framework for forecasting PV system performance.

Overcoming the drawbacks of exclusively data-driven or physics-based methods is one of the primary drivers behind the use of hybrid models, with physical models being less adaptable for handling uncertainties in the actual world by demanding precise system parameters and high-quality meteorological data, despite being dependable for estimating long-term energy yield. Conversely, machine learning models are excellent at identifying intricate patterns in data, but they frequently falter when used in unfamiliar situations or with sparse training data, potentially improving predictive skills with adaptive learning strategies and gaining from the correctness of physical equations.

Creating artificial training datasets for machine learning algorithms using physical models is a popular hybrid modeling technique, with artificial neural networks (ANNs), support vector machines (SVMs), and gradient boosting techniques being examples of artificial intelligence models that can be trained using simulated data from physics-based models when historical PV system data is unavailable, making it possible for machine learning models to pick up basic system characteristics while still adjusting to changes in the actual world [34].

Model fusion is another hybrid approach that combines the results of machine learning with physical models to increase prediction accuracy, with a machine learning model correcting discrepancies by taking into account environmental disturbances like cloud cover or soiling losses, whereas a physical irradiance model estimating the amount of solar energy that is available [35]. Similar to this, by dynamically modifying power loss estimations in response to current weather conditions, deep learning approaches can improve the temperature correction models utilized in physical simulations.

In applications like fault detection, performance improvement, and short-term PV forecasts, hybrid models are especially useful, with physical models, for instance, giving baseline projections for energy production forecasting, while machine learning algorithms improve these estimates by analyzing past variances, being particularly helpful in grid-connected photovoltaic systems, where demand-supply balance and power management depend on precise short-term forecasts.

Despite their benefits, hybrid models need to be carefully designed to guarantee computational efficiency and interpretability, as it takes domain expertise to integrate physics-based equations with machine learning, and too complex models could result in higher computing costs without appreciable accuracy gains, with hybrid models however emerging as a potent instrument for enhancing PV system performance monitoring and energy forecasting as artificial intelligence and solar modeling continue to progress, making them an attractive strategy for both research and commercial applications.

### 2.4.4 Simulation Tools (pvlib, PVsyst, SAM, HOMER, MATLAB-Based Tools)

Because simulation tools offer in-depth evaluations of energy production, efficiency, and financial viability, they are essential for modeling and improving the performance of photovoltaic systems, estimating solar energy generation under various operational and environmental conditions by combining physical models, empirical data, and sophisticated algorithms. Numerous software tools are available, each with unique features suited to feasibility studies, system design, and research, depending on the application [36].

An open-source Python package called pvlib is frequently used to model the performance of PV systems in both industry and academic settings, offering features for estimating power output based on meteorological data, modeling irradiance, and calculating solar position, with one of its main advantages being its adaptability, which enables users to incorporate unique methods and change modeling parameters, making it very helpful for projects involving complex system analysis and research.

A commercial software program called PVsyst was created to simulate PV systems in great detail, having the ability to mimic several PV installation types in detail, including hybrid, off-grid, and grid-connected systems. Engineers and consultants working on PV system design and feasibility studies use PVsyst as it integrates loss factor computations, shading analysis, and meteorological information, being also useful for large-scale solar projects because it can perform financial analysis and simulate various inverter designs.

Another effective tool for PV system performance and financial modeling is the System Advisor Model (SAM), which was created by the National Renewable Energy Laboratory (NREL) of the U.S. Department of Energy. Through a variety of system configurations and economic situations, such as incentives, tax credits, and financing alternatives, SAM enables users to assess energy output, resulting in it being especially advantageous for investors, legislators, and project developers who must evaluate the long-term feasibility of solar energy projects.

A specific program called HOMER (Hybrid Optimization of Multiple Energy Resources) was created to optimize hybrid energy systems that consist of diesel generators, PV, wind, and battery storage, an by its mimicking energy output, storage, and consumption under various load circumstances, it is very helpful for microgrid and off-grid applications, conducting economic and sensitivity studies in order to assist decision-makers in optimizing system configurations based on technical performance and cost-effectiveness [37]. Simulink and PV-specific toolboxes are two examples of MATLAB-based solutions that offer highly configurable settings for simulating and controlling PV systems, with advanced algorithms for energy forecasting, optimization, and real-time system monitoring being commonly developed using MATLAB, where its adaptability makes it a popular option for research and development, especially when it comes to incorporating machine learning and artificial intelligence methods into PV modeling.

Since each of these tools has unique advantages, they can be used for various PV system analysis tasks, with PVsyst and SAM providing organized frameworks for system design and financial assessment, while MATLAB-based and pvlib-based tools offer flexibility for research applications, where HOMER is in turn perfect for optimizing hybrid energy. The particular goals of the study—whether they include energy predictions, financial viability analysis, thorough physics-based simulations, or hybrid system integration—will determine which method is best, with the accuracy and efficiency of solar energy modeling significantly improving as PV technology develops by combining these simulation tools with big data analytics and artificial intelligence.

## 2.5 Optimization Techniques in Renewable Energy

### 2.5.1 Evolutionary Algorithms (Genetic Algorithms, Particle Swarm Optimization, DEAP)

Natural selection and collective intelligence serve as the inspiration for evolutionary algorithms (EAs), a family of optimization techniques that are frequently used in renewable energy optimization, particularly improving the performance of photovoltaic (PV) systems, being especially good at solving complicated, multi-dimensional optimization problems where conventional mathematical methods could falter because of their high computational complexity and non-linearity [38]. Genetic methods (GA), Particle Swarm Optimization (PSO), and implementations utilizing the DEAP (Distributed Evolutionary Algorithms in Python) library are some of the most often utilized evolutionary methods in PV optimization.

Darwinian evolution serves as the inspiration for Genetic Algorithms (GA), which iteratively evolve a population of candidate solutions toward an ideal answer through mechanisms like crossover, mutation, and selection, being commonly used in PV systems to optimize system design parameters such panel tilt angles, inverter configurations, and battery storage sizing, in order to improve energy yield and save costs [39]. It has the advantage of efficiently exploring vast search spaces without the need for gradient information, making it appropriate for situations involving several local optima, with its convergence performance potentially being rather slow however, especially when working with extremely complicated systems.

In order to identify the best solutions, Particle Swarm Optimization (PSO) uses the collective behavior of swarms, such as fish schools or flocks of birds [40]. Using their own expertise and the best-known solutions discovered by other particles, candidate solutions (particles) navigate the search space in PSO, being effectively used in PV system optimization for tasks like fine-tuning control parameters for hybrid renewable energy systems, optimizing Maximum Power Point Tracking (MPPT) algorithms, and maximizing power production under fluctuating weather situations. PSO tends to converge more quickly and requires fewer tuning parameters than GA, but if the search space is extremely irregular, thus occasionally becoming stuck in local optima.

An open-source Python package called DEAP (Distributed Evolutionary Algorithms in Python) was created to effectively implement evolutionary algorithms, offering an adaptable framework for creating unique optimization plans utilizing PSO, GA, and other heuristic techniques [41]. Because it enables researchers to combine evolutionary methodologies with machine learning models, DEAP is especially helpful in PV system optimization, allowing for intelligent energy management and adaptive energy forecasting. Large-scale optimization challenges, such constructing PV-powered microgrids or optimizing multi-objective functions that simultaneously take efficiency, cost, and environmental effect into account, can also benefit from its distributed computing capabilities, and in situations where it is necessary to balance several competing goals, evolutionary algorithms have shown themselves to be effective tools for optimizing renewable energy systems. The particular problem, available computing power, and the necessary balance between exploration (finding new solutions) and exploitation (improving on existing ones) all influence the decision between GA, PSO, or DEAP-based techniques, with the efficiency and adaptability of PV system optimization being anticipated to be significantly improved as the area of renewable energy develops through the integration of evolutionary algorithms with real-time data analytics and artificial intelligence.

### 2.5.2 Mathematical Optimization Methods (Linear/Non-linear Programming, Convex Optimization)

Because mathematical optimization techniques offer organized and effective ways to address energy-related optimization issues, they are essential for improving the performance of photovoltaic systems, relying on precisely defined mathematical formulations and limitations to identify the best solutions for PV system design, operation, and energy management, as opposed to heuristic or evolutionary algorithms, with convex optimization, linear programming (LP), and non-linear programming (NLP) being some of the most widely utilized techniques, and their application depending on the type of problem and its complexity [42].

One of the most straightforward and popular optimization methods for managing energy systems is linear programming, or LP, being applied when linear equations may be utilized to define the goal function and restrictions and being frequently used in PV system contexts for energy dispatch issues, battery charge-discharge timing, and cost-cutting tactics. For instance, LP can be used to identify the best combination of grid power, solar energy, and battery storage to reduce electricity prices while meeting energy demand, and when the problem retains linearity, LP's efficiency is found in its capacity to deliver quick and globally optimal solutions, with complicated interactions between variables however being present in many real-world PV optimization issues, making it impossible to correctly characterize them with linear equations alone.

By adding non-linear correlations between decision variables, Non-Linear Programming expands on linear programming and offers a more adaptable method for optimizing photovoltaic systems, being necessary for effective modeling of many non-linear phenomena in PV energy production, such as the power output of solar panels in response to temperature and irradiance changes. NLP is very helpful for improving hybrid renewable energy system setups, inverter control techniques, and MPPT algorithms, and although it can produce more realistic results than LP, it requires more computing power and may have trouble guaranteeing global optimality because non-linear problems can have several local optima.

Convex optimization is a subfield in natural language processing that focuses on situations in which the constraints and objective function form a convex set, and since its techniques provide robust theoretical assurances of effectively locating the global optimum, they are frequently chosen in PV system applications, with convex optimization problems potentially being used to define a variety of energy system optimization issues, including demand-side energy management, the economic dispatch of renewable energy sources, and power flow optimization in microgrids. Within this context, methods like second-order cone programming and quadratic programming are frequently employed, having the advantage of producing high-precision solutions when using reputable numerical solvers, being only applicable to issues that can be reformulated to preserve convexity, which isn't necessarily feasible for interactions in extremely complicated PV systems.

### 2.5.3 Machine Learning-Based Optimization (Neural Networks, Reinforcement Learning)

An increasingly potent technique for optimizing renewable energy systems, such as photovoltaic (PV) installations, is machine learning (ML), with extensive computer resources and thorough system modeling being frequently needed for traditional optimization techniques like evolutionary algorithms and mathematical programming. On the other hand, ML-based optimization techniques are especially well-suited for intricate and dynamic PV energy systems as they can recognize patterns in past data and adjust to shifting circumstances, with neural networks and reinforcement learning being two of the most used machine learning approaches in PV system optimization.

**Neural Networks for PV Optimization**

The ability of artificial neural networks (ANNs) to represent intricate, non-linear interactions between input variables and outputs makes them popular in applications involving renewable energy, as they can be taught to forecast energy production, maximize power output, and enhance MPPT performance in a variety of environmental settings in PV system optimization. In order to optimize efficiency, a neural network, for instance, can understand the relationship between temperature, power output, and solar irradiance and make modifications in real time. Furthermore, grid stability analysis, PV array defect detection, and energy forecasting have all made use of deep learning architectures like convolutional and recurrent neural networks, with neural networks' capacity to generalize from past data and continuously enhance predictions over time being their primary benefit, requiring however, to be trained on sizable datasets and are susceptible to overfitting if improperly regularized [43].

**Reinforcement Learning for PV System Optimization**

A subfield of machine learning called reinforcement learning (RL) uses trial and error in a dynamic environment to teach an agent how to make the best choices, with RL algorithms having demonstrated significant promise, especially in fields like energy management, battery storage optimization, and grid interface tactics in PV system optimization [44]. Without the requirement for explicit system modeling, RL can learn and adjust to shifting energy demands, weather patterns, and electrical market circumstances, in contrast to conventional rule-based techniques, with an RL-based controller, for example, balancing energy availability and cost savings by optimizing a battery's charging and discharging schedule in a hybrid PV system. RL agents gradually enhance their decision-making skills by interacting with the environment on a constant basis and earning incentives based on performance measures (such as cost reduction or efficiency), with two of the main obstacles to RL in PV optimization being the requirement for substantial training and processing resources, together with the possibility of less-than-ideal actions during the learning phase. In PV systems, machine learning-based optimization offers a potent substitute for conventional optimization methods, facilitating flexible, data-driven decision-making that raises cost-effectiveness and energy efficiency, with PV systems functioning more intelligently and independently by utilizing reinforcement learning for dynamic optimization and neural networks for predictive modeling.

### 2.5.4 Multi-Objective Optimization for PV Performance (Efficiency, Cost, Storage Integration)

Maximizing energy efficiency, cutting costs, and incorporating energy storage technologies are just a few of the many, frequently incompatible goals that must be balanced when optimizing photovoltaic (PV) systems, with a framework for handling these trade-offs being offered by multi-objective optimization (MOO) approaches, guaranteeing that PV systems function at their best while still being financially feasible [45]. MOO looks for optimal solutions that concurrently meet several criteria, as opposed to single-objective optimization, which concentrates on a single statistic, enabling decision-makers to select the optimum trade-off from these alternatives based on their unique goals and restrictions, which are sometimes depicted as a Pareto front.

**Efficiency Optimization in PV Systems**

One of the main goals in the design and operation of PV systems is to maximize energy efficiency, with optimized panel orientation, sophisticated MPPT algorithms, and system-level modifications like charge controller and inverter tuning all increasing efficiency. Efficiency maximization is frequently combined with additional considerations, like durability and maintenance costs, in multi-objective techniques, and to ensure long-term performance stability, an optimization algorithm can, for instance, aim to optimize energy yield while minimizing PV panel thermal depreciation, with such multi-objective issues being frequently solved using machine learning and metaheuristic algorithms, like Genetic Algorithms and Particle Swarm Optimization, dynamically determining the optimal operational parameters.

**Cost Minimization in PV Deployment**

For PV systems to become more widely used and economically viable, their total cost must be decreased, with the reduction of capital expenditures (CAPEX), operating costs (OPEX), and levelized cost of energy (LCOE) being all components of cost optimization [46]. Different PV system topologies can be assessed using multi-objective optimization models, which balance long-term energy savings with early investment costs, with an MOO algorithm, for instance, possibly figuring out how many solar panels, inverters, and storage units are best to have in order to maximize ROI and guarantee sufficient energy production. To increase the financial appeal of PV systems, cost-based optimization frameworks can also incorporate government regulations, financial incentives, and energy pricing models.

**Energy Storage Integration for PV Systems**

For PV systems to be more energy reliable, especially in off-grid and hybrid applications, battery storage must be integrated, having to balance variables like battery longevity, charging/discharging cycles, and energy availability; however, they complicate the optimization process. Finding the ideal battery type (such as lithium-ion vs. lead-acid), storage capacity, and operational plan all depend heavily on multi-objective optimization, with a well-optimized system needing to make sure that extra solar energy is effectively stored and used strategically to lessen dependency on the grid or backup generators. For example, depending on projections of solar power, electricity demand, and market prices for grid-connected systems, optimization models can identify the optimal battery charging schedule.

## 2.6 Forecasting Energy Consumption: Methods and Challenges

### 2.6.1 Statistical Methods (ARIMA, Exponential Smoothing)

Because statistical approaches are efficient, interpretable, and can efficiently model time series data, they have been used for a long time to forecast energy use, with the AutoRegressive Integrated Moving Average (ARIMA) model and exponential smoothing techniques being two of the most popular methods for forecasting future energy demand; both rely on past consumption trends, being appropriate for short- to medium-term forecasting in photovoltaic (PV) systems and other energy-related applications as they work especially well when energy consumption exhibits distinct trends and seasonal patterns [47].

**AutoRegressive Integrated Moving Average (ARIMA)**

One of the most well-known statistical methods for time series forecasting is the ARIMA model, consisting of three main parts: Moving Average (MA) terms, using previous forecast errors to improve predictions; AutoRegressive (AR) terms, using past values to predict future values; and Integrated (I) terms, taking trends into account by differencing the data. With p standing for the number of lag observations, d for the degree of differencing, and q for the number of moving average terms, the ARIMA model is commonly written as ARIMA(p, d, q), simulating demand swings for energy consumption forecasts by incorporating seasonal variations (SARIMA) and outside factors like temperature and economic activity. Its performance may be constrained when dealing with extremely variable or nonlinear patterns of energy consumption, though, because it makes the assumption that relationships between past and future values would remain constant across time, being still a powerful baseline model and being frequently used to compare more intricate machine learning techniques despite this drawback.

**Exponential Smoothing Methods**

Another class of statistical forecasting approaches that give historical observations exponentially decreasing weights are exponential smoothing methods, being especially helpful for identifying short-term trends in energy usage since it guarantees that more current data will have a bigger impact on the forecast, with different forms of exponential smoothing existing, including the following:

* Data without notable trends or seasonality can be forecasted using simple exponential smoothing, or SES, with the amount of weight assigned to recent observations being controlled by a single smoothing parameter (α).
* By adding a trend component, Holt's Linear Trend Model expands on SES and improves its performance for energy consumption data that shows growth or fall over time.
* By including a seasonal component, Holt-Winters Exponential Smoothing enhances Holt's approach even more and is especially helpful for PV-related energy forecasts, where demand and production vary seasonally.

Exponential smoothing models are appealing for real-time energy forecasting applications because they are computationally efficient and require fewer parameters than ARIMA, finding it difficult to identify intricate trends or abrupt shifts in energy usage however, because of outside influences like shifting regulations, severe weather, or advances in technology.

### 2.6.2 Machine Learning Models (ANNs, LSTMs, Hybrid Approaches)

Μachine learning algorithms have attracted a lot of attention in the field of energy consumption predictions being able to handle complicated, nonlinear patterns and adjust to changing situations, learning from enormous datasets and uncovering complex correlations between weather, energy consumption, and other external factors, in contrast to classic statistical models that rely on predefined mathematical relationships. Artificial Neural Networks and Long Short-Term Memory (LSTM) networks are two of the most popular machine learning models for energy forecasting, with hybrid approaches, on the other hand, integrating several techniques to improve prediction accuracy.

**Artificial Neural Networks (ANNs)**

Artificial Neural Networks (ANNs) are computer models made of interconnected layers of artificial neurons that are modeled after the structure of the human brain, being extremely good at capturing nonlinear interactions, making them ideal for forecasting energy consumption in situations where demand variations are influenced by a number of interconnected variables, including temperature, time of day, and economic activity. ANNs are usually composed of an input layer that receives external factors and historical energy usage, hidden layers that perform calculations and recognize patterns, and an output layer that generates the final forecast, continuously modifying the weights of links through a procedure known as backpropagation to reduce forecasting mistakes [48]. The ability of ANNs to handle a variety of datasets is one of its main advantages; yet, for efficient training, they need a lot of data and processing power, while possibly overfitting if improperly regularized, meaning that while they may perform well on training data, they may not generalize to new data.

**Long Short-Term Memory (LSTMs) Networks**

Time series forecasting benefits greatly from the usage of LSTMs, a specific kind of Recurrent Neural Network (RNN) made to process sequential data, contrasting to conventional feedforward networks, address the short-term dependencies present in conventional RNNs by preserving memory of previous observations over extended periods of time, being excellent at capturing temporal dependencies, such as daily, weekly, or monthly patterns, in energy consumption forecasting. The gating mechanism, which selectively remembers or forgets information, is the main novelty of LSTMs, enabling the model to ignore noise and concentrate on pertinent previous data, resulting in them being very good at forecasting energy use in settings with high temporal correlations. But like ANNs, LSTMs need a lot of training data and processing power, and they're still difficult to comprehend when compared to more conventional statistical techniques.

**Hybrid Approaches**

Hybrid models integrate statistical, machine learning, and deep learning techniques to enhance predictive performance by utilizing the advantages of several forecasting approaches, and while ANNs or LSTMs are used to understand nonlinear relationships, ARIMA or exponential smoothing models are frequently used to capture linear and seasonal trends. Compared to standalone models, hybrid models can produce projections that are more accurate and consistent by combining several techniques, with a hybrid ARIMA-LSTM model for example, eliminating seasonality and trends from the dataset using ARIMA first, and then sending the residual errors to an LSTM network to learn intricate patterns. Similar to this, energy consumption data may be broken down into several frequency components using wavelet transformation techniques, which deep learning models can subsequently interpret, ensuring improved generalization using optimization techniques or genetic algorithms to fine-tune machine learning model parameters. Although hybrid models increase computing complexity, they have shown greater forecasting accuracy in a variety of renewable energy applications, with a number of variables, including predicting time horizon, processing resources, and data availability, influencing the choice of suitable hybridization components.

### 2.6.3 Data Requirements and Preprocessing (Time Series Data, Weather Dependencies)

The quality and accessibility of data are critical components of accurate energy consumption forecasts, with the energy demand showing complex patterns that are impacted by a number of outside variables, therefore accurate forecasts depend on appropriate data collection, cleaning, and preprocessing, as the time series data and weather dependencies are the two most important data elements in energy forecasting, they need to be properly controlled before using statistical or machine learning models.

**Time Series Data**

Since observations are gathered successively throughout time, energy consumption data is by its very nature a time series, requiring preprocessing to take into consideration the seasonal patterns, trends, and cyclical fluctuations that are frequently present in time series data. For instance, the demand for power usually fluctuates on a daily and weekly basis, peaking during business hours and falling off at night, with long-term trends potentially also having a big impact on patterns of energy consumption, including the growing use of renewable energy sources or changes in industrial activity, thus requiring several preparation processes to get time series data ready for forecasting.

* Firstly, it is essential to manage missing data, as sensor malfunctions, misunderstandings, or maintenance intervals can all result in data gaps, with common methods to deal with missing values being forward or backward filling, statistical models to estimate missing points, and interpolation (spline, linear).
* Secondly, it is equally important to find and eliminate outliers, as forecasting models may become distorted by unusual increases or decreases in energy usage. Equipment failures, power outages, or anomalous human activity can all lead to outliers, with machine learning-based anomaly identification, and Z-score analysis can find and fix such anomalies with the aid of techniques like box plots.
* Thirdly, there must be normalization and scaling of input features to a normal range, enabling many forecasting models—especially machine learning algorithms—to perform better, with standardization (z-score normalization) or min-max scaling being frequently employed to guarantee that every variable makes a significant contribution to model training.
* Finally, there must be feature engineering, enhancing model performance by adding more time-based features, such as the day of the week, the hour of the day, holidays, or lagged energy consumption values (autocorrelations), with the most pertinent predictors being found using feature selection techniques like mutual information scores and principal component analysis (PCA).

**Weather Dependencies**

Weather patterns have a big impact on how much energy is used, especially in systems that use renewable energy sources like wind and solar, with the demand for and production of electricity being directly impacted by variables such as temperature, humidity, solar radiation, wind speed, and cloud cover. For example, solar panel efficiency is dependent on ambient temperature and sunshine availability, while extreme temperatures lead to higher energy demand for heating and cooling, thus suggesting the following actions as necessary in order to include weather dependencies in forecasting models [49].

* Integration of Meteorological Data from satellites, internet APIs, or weather stations, frequently used for energy forecasting models, being synchronized with energy usage records to guarantee precise correlations.
* Weather Forecasting for Future Predictions, as accurate short- and long-term weather forecasts are required as model inputs since energy demand forecasting frequently goes into the future, being produced using methods like machine learning-based weather forecasting and numerical weather prediction (NWP) models.
* Finally, Managing Spatial Variability, as weather patterns might differ from one place to another in regional energy forecasts, thus potentially requiring geospatial methods or interpolated meteorological data in order to account for regional variations, if a forecasting model includes several locations.

### 2.6.4 Challenges and Future Directions

The creation of highly accurate and dependable predictive models is hampered by a number of issues that persist despite notable progress in energy consumption forecasts, stemming from the complexity of patterns in energy demand, the quality and availability of data, outside influences, and the shortcomings of current forecasting methods. In order to overcome these challenges, ongoing research and innovation are needed, requiring future directions to concentrate on using large data, incorporating sophisticated machine learning models, and enhancing real-time adaptability.

**Challenges in Energy Consumption Forecasting**

Data ambiguity and unpredictability are major obstacles to energy consumption predictions, with numerous factors, such as consumer behavior, economic activity, technical breakthroughs, and unforeseen occurrences like pandemics or natural disasters, all having an impact on energy consumption, contributing into the data while traditional forecasting models find it challenging to sustain steady accuracy over time because to the non-linear and non-stationary features.

The availability and quality of data is another significant problem, with effective forecasting models requiring high-quality datasets for training, but real-world energy consumption data frequently has outliers, missing values, and inconsistencies. Furthermore, it is difficult to create reliable models since many areas or industries lack comprehensive historical records, and integrating weather data—which is essential for precise forecasting—presents additional challenges due to temporal and regional fluctuations in meteorological conditions.

There are further difficulties with model interpretability and complexity, making it challenging to understand their judgments, as even deep learning models like long short-term memory networks and artificial neural networks have shown excellent prediction performance frequently operating as "black boxes,", impeding confidence and acceptance, especially in sectors like energy markets and grid management where decision-making must be explicable in real-world applications. Furthermore, scalability and computing efficiency continue to be issues, particularly in the context of smart metering systems and large-scale energy grids, with large-scale deployment and real-time forecasting being difficult due to the high computational resource requirements of many sophisticated machine learning techniques, making forecasting more difficult, as renewable energy sources like solar and wind power are erratic and intermittent.

**Future Directions in Energy Forecasting**

Future energy forecasting research is probably going to concentrate on a few important areas in order to overcome these issues, with the extensive use of sensors, smart meters, and grid monitoring equipment producing enormous volumes of real-time data that can be used to improve forecasting precision, and then processed more effectively with the use of cloud and edge computing systems, allowing for adaptive forecasting models and forecasts that are almost instantaneous. The use of hybrid forecasting models, which integrate several approaches like machine learning, statistical methodologies, and physical models, is another significant breakthrough, enhancing precision, flexibility, and resilience to uncertainty by utilizing the advantages of many methodologies, potentially adapting dynamically in response to real-time data inputs by combining reinforcement learning with conventional time series forecasting techniques.

Energy forecasting is also anticipated to be significantly impacted by explainable AI (XAI) techniques in the future, helping stakeholders, legislators, and energy suppliers comprehend the fundamental elements affecting demand forecasts, with complex forecasting algorithms made more transparent with the use of techniques like SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations). Furthermore, by making it possible to handle complicated datasets more quickly and effectively, developments in quantum and neuromorphic computing have the potential to completely transform energy forecasting, increasing long-term forecasting models' accuracy and decreasing calculation times drastically.

Lastly, the future of energy forecasting will be greatly influenced by legislative and regulatory assistance, with investments in data-sharing programs, energy reporting standards, and the creation of AI-powered grid management systems being imperative for governments and energy organizations, while at the same time, promoting cooperation among academic institutions, business, and policymakers can accelerate the uptake of cutting-edge forecasting methods and improve the general stability of energy systems.

# CHAPTER 3 Methodology

The methodological framework used to simulate, optimize, and forecast the energy consumption and performance of photovoltaic (PV) systems is described in this chapter, combining theoretical and data-driven approaches, modeling tools, optimization algorithms, and machine learning models in order to assess and improve the performance of PV systems under various operating and environmental circumstances. Accurately simulating the energy production of PV systems, adjusting system parameters to increase efficiency and cost-effectiveness, and anticipating energy consumption patterns for improved energy management are the main goals of the methodology, with every element of the approach being chosen to be in line with the practical needs of contemporary PV systems as well as the limitations of technology.

There are three primary sections to the chapter, with the simulation of PV systems being the main topic of the first section, also describing the assumptions, input parameters, and techniques used to predict energy generation in realistic settings. The optimization methods used to improve PV system performance, including parameter tweaking and storage solution integration, are presented in the second section, focusing on data preprocessing, model selection, and performance evaluation, and finally concluding by outlining the forecasting methodology used to estimate future energy demand.

## 3.1 Core Research Questions

Optimizing photovoltaic systems to better match building consumption patterns is one of the main research concerns guiding this work, aiming to cover both the technical and financial elements of PV system performance. Optimization is an important problem in raising the effectiveness and financial sustainability of solar energy systems.

The first question is “How may PV systems be adjusted to better fit the consumption patterns of buildings?” looking at how PV systems may be optimized to better fit the building's energy usage profile, with PV energy generation being intrinsically variable, output varying during the day according to weather, season, and solar irradiation. On the other hand, the building's utilization, which might change depending on the time of day, season, and occupancy, affects patterns of energy consumption. Thus the study aims to determine methods for coordinating energy production and consumption in order to lessen dependency on the grid and boost system efficiency by examining these trends, with the integration of energy storage, the function of energy management systems in optimizing load distribution, and the size of the PV system in relation to consumption requirements being important factors.

The second study question is “Which orientation, tilt or azimuth, is best for balancing production and consumption?”, exploring the connection between solar panels' tilt angle and azimuth and their capacity to balance building energy consumption and energy output, with PV panel exposure to sunlight, which is impacted by the panels' tilt and orientation, having a significant impact on the panels' efficiency, with the optimal panel orientation in residential and commercial settings potentially changing based on a number of variables, including geographic location, roof design, and patterns of energy demand throughout the day. In order to increase self-consumption and decrease dependency on external power sources, this inquiry seeks to determine the best panel arrangement that maximizes energy output during times when the building is most in need of electricity, taking into account local weather patterns, seasonal variations, and the possible advantages of mounting methods that may be adjusted.

The third question is “How much battery capacity is necessary to attain the best possible self-consumption?”, making sure that the energy generated during the hours of greatest sunlight is usable when demand is high, especially during times of low or no sunlight, thus optimizing the self-consumption of solar energy. Finding the right battery capacity needed to reach this ideal degree of self-consumption is the main goal of the third research topic, with the size of energy storage devices, which store extra solar energy produced during the day for use at night or during overcast conditions, being the subject of this question. Estimating the required battery capacity will take into account a number of factors, including the building's energy demand profile, the PV system's capacity, and the state of storage technology, while also evaluating the financial effects of purchasing larger storage systems as well as how these systems affect the PV installation's long-term efficacy and cost-effectiveness.

The final question is “What are this approach's technical, seasonal, and financial ramifications?”, examining in particular, the technical, seasonal, and economic aspects of optimizing PV systems for better alignment with building consumption patterns. Technically speaking, the study will look at how various system elements, like charge controllers, batteries, and inverters, can be combined to produce the best possible system performance, focusing on how system optimization techniques might be modified to satisfy shifting energy demands in various seasons, and taking into account how energy production and consumption fluctuate throughout the year. Lastly, the economic implications will concentrate on the cost-benefit analysis of these optimization tactics, including the payback periods for various optimization methodologies, the return on investment for energy storage systems, and the possible energy bill savings, aiming to give a thorough grasp of the advantages and disadvantages of optimizing PV systems for energy independence and self-consumption.

## 3.2 Methodological Framework

An integrated multi-dimensional optimization framework has been created in order to successfully optimize photovoltaic systems for building consumption and answer the main research objectives, combining a number of approaches and resources intended to assess and enhance system performance in a comprehensive manner. To get the best possible energy production, consumption alignment, and economic efficiency, the strategy integrates multi-objective optimization approaches, intricate physical modeling, and a thorough examination of seasonal and temporal dynamics.

Multi-objective genetic algorithms (MOGAs) are used to simultaneously maximize a number of conflicting goals, including maximizing energy self-consumption, lowering expenses, and guaranteeing system dependability, with natural selection serving as the inspiration for genetic algorithms, which are especially well-suited for complicated optimization problems involving a lot of variables and non-linear connections, like PV system optimization, with system size, battery capacity, inverter settings, panel orientation, and other design characteristics all being taken into account at the same time using the multi-objective approach, which also determines the optimum trade-offs between them. Decision-makers can choose the configuration that best fits their unique requirements and objectives thanks to the solution set that is produced, which represents the most ideal configurations.

An important part of the framework is also the comprehensive physical modeling of the PV system, supporting the optimization process, simulating the energy generation of the solar panels using location-specific meteorological and irradiance data, together with system factors like panel efficiency, tilt, and orientation. When taking into account variables like temperature fluctuations, shading effects, and panel deterioration over time, physical models offer a realistic depiction of how solar energy is captured, transformed, and stored. Predicting the PV system's energy output and comprehending how it will function in various environmental circumstances depend heavily on accurate physical modeling, making sure that the outcomes are both practically and theoretically feasible by combining it with the optimization techniques.

Aligning energy output and consumption is a major difficulty in developing PV system optimization, with the framework placing particular emphasis on temporal consumption-production alignment, entailing coordinating solar energy generation with the energy demand trends of the building. The optimization model modifies the energy production strategy by examining past consumption data to guarantee that the building will have access to the electricity produced during the peak sun hours when it is most needed, promoting self-consumption and lessens dependency on the grid, while also modifying the system's functioning to account for the fluctuating energy demand throughout the day, which may include larger loads in the morning or evening.

Seasonal performance analysis is a crucial component of the optimization process since solar energy production varies with the seasons, and taking into account variables like sun angle, day length, and climate variations, it assesses how well the PV system will function throughout the year. The fact that solar panel energy production varies throughout the year, often being higher in the summer and lower in the winter, makes this research important, assisting in figuring out how to modify system parameters, like tilt angle or battery usage, to ensure constant performance throughout the year by looking at seasonal patterns. Even when there is less sunlight, the PV system remains dependable and efficient thanks to its seasonal modification.

Determining the proper battery capacity needed to store extra solar energy is an important step in improving energy self-consumption, with battery requirements modeling, which models storage requirements based on daily energy use patterns and solar production estimates, being incorporated into the framework. The system can function independently of the grid for extended periods of time by precisely simulating the storage capacity required to guarantee that excess energy generated during the day may be used throughout the night or during cloudy conditions, assisting in determining the ideal battery size by taking into account both the necessary autonomy and the available budget.

Finally, the cost-effectiveness of the optimized PV system configurations is assessed by the thorough economic assessment, with the original capital investment, continuing maintenance expenses, energy savings, and other financial incentives like tax credits or government subsidies all being taken into consideration in this assessment. Metrics such as the payback period, ROI, and levelized cost of energy (LCOE) are used in the economic evaluation to determine whether the optimized PV systems are financially viable, with the framework guaranteeing that the final recommendations are not only technically and environmentally efficient but also financially viable by incorporating economic factors into the optimization process, thus giving stakeholders practical insights into the economic implications of various system configurations.

## 3.3 Multi-Objective Optimization Approach

To address the multi-objective optimization problem at hand, we used a proven evolutionary technique, the Non-dominated Sorting Genetic technique II (NSGA-II), being chosen in particular because of how well it optimizes multifaceted, intricate issues with conflicting goals. With this strategy, we were able to strike a compromise between the system's two main objectives: optimizing overall energy production and reducing the weighted energy mismatch between production and consumption. Iterating through a vast population of possible solutions and comparing each one to the two specified objective functions comprised the NSGA-II optimization process, providing a set of best solutions known as the Pareto front by ranking and sorting the answers according to their dominance relationship by using a Pareto-based methodology, enabling the selection of a configuration that best satisfies the desired criteria by representing various trade-offs between the conflicting objectives.

### 3.3.1 Objective Functions

We established the following objective functions in order to measure the optimization's two main goals:

**Reduce the Weighted Energy Disparity**

1. In order to ensure that the system produces enough power to meet consumption requirements while reducing reliance on the grid, the first goal function seeks to reduce the energy mismatch between the solar power produced and the energy required by the building, with the following being a mathematical expression for the energy mismatch:
   * θ is the tilt angle of the solar panels
   * φ is the azimuth angle of the solar panels
   * P(θ,φ,i) is the power production at hour i, based on the tilt and azimuth angles
   * C(i) is the energy consumption at hour i
   * wi is the weighting factor for hour i, which adjusts the importance of energy mismatch at different times of the day
   * N is the total number of hours considered in the optimization (typically over a full day or seasonal cycle)

**Optimize the Production of All Energy**

1. The goal of the second objective function is to maximize the PV system's overall energy production during the optimization period, being especially important for making sure the system produces the most solar energy possible, which can either be stored for later use or used for self-consumption, and the following equation illustrating how the energy production is totaled over all time periods:

Where:

* P(θ,φ,i) is the power generation from tilt and azimuth angles at hour I
* N is the total number of hours taken into account during optimization.

In order to maximize the use of solar energy and minimize the demand for grid electricity, this function makes sure that the system's energy generation is maximized, with the system further improving energy independence by ensuring that extra energy is accessible for storage by optimizing energy production.

**Optimality and Pareto trade-offs**

Due to the trade-off analysis between these two goals conducted by the NSGA-II algorithm, an ideal solution might not always fully meet both goals, with the algorithm rather finding a set of Pareto-optimal solutions, meaning that no other option can enhance one goal without making the other worse, balancing the trade-off between optimizing energy production and avoiding energy mismatch. For instance, a solution that maximizes production may result in a greater mismatch between production and consumption, whereas a solution that achieves a low energy mismatch may result in somewhat less energy production, with decision-makers being able to select from a range of configurations provided by the optimization process, depending on their goals, such as whether they prioritize increased energy output or higher self-consumption.

### 3.3.2 Genetic Algorithm Implementation

To find the ideal tilt and azimuth angles that balance the goals of minimizing energy mismatch and maximizing energy production, we used a Genetic Algorithm (GA) in our optimization process, and especially, a Non-dominated Sorting Genetic Algorithm II (NSGA-II), effectively managing multi-objective optimization problems, and the main elements of the GA implementation being listed below:

* A chromosome made up of the tilt angle (θ) and azimuth angle (φ) genes represent each individual in the population, specifying the solar panels' orientation, which has a direct impact on how well they match energy production and consumption. The tilt and azimuth values that the GA develops across generations are thus associated with each solution to the optimization issue.

**Chromosome: [tilt angle,azimuth angle]**

* Due to the physical constraints and ideal placement of the solar panels, the tilt and azimuth angles are limited to particular ranges, with a level panel being represented by a tilt angle (θ) of 0°, while a vertically inclined panel is represented by a 90° tilt angle. The azimuth angle (φ) is configured to cover orientations from east to west and vary from 90° to 270°, guaranteeing that the panels can face any direction using this range.

Tilt: 0°≤θ≤90°

Azimuth: 90°≤φ≤270°

* The number of potential solutions (individuals) in each generation is referred to as the population size, and depending on the intricacy and level of precision needed for the optimization work, it is adjusted to range from 15 to 50 individuals, with a bigger population size providing for increased diversity in the search space despite increases computing work.
* Individuals were chosen for reproduction using the tournament selection procedure in conjunction with NSGA-II rating, with a number of people chosen at random, and the most fit person among them is selected to help the next generation during the tournament selection process. By further rating solutions according to Pareto dominance, NSGA-II ranking makes sure that those that perform better on both objective functions are given preference.
* We employed a blend crossover with an alpha parameter (α) of 0.5 to merge the genetic material of two parent solutions, combining the genes (tilt and azimuth angles) of two parent solutions to produce children. While preserving some continuity with the parents, the blend crossover operator guarantees that children are generated inside the possible search space and promotes the investigation of various solutions.
* A Gaussian mutation operator was used with a standard deviation (σ) of 5 and a mean (μ) of 0, enabling the algorithm to better explore the search space and avoid local optima, thus adding tiny, random fluctuations to the tilt and azimuth angles. In order to prevent significant changes to solutions, the Gaussian distribution makes sure that the mutations are concentrated around the current values.
* During the optimization phase, the mutation probability was modified to enhance convergence and avoid premature stagnation, and as the program developed, the mutation probability was progressively decreased from its initial value of 0.2 to 0.05, encouraging exploitation as the algorithm gets closer to convergence and exploration early in the optimization process.
* Finally, until one of two requirements was satisfied, the optimization process kept going:
  1. Usually ranging from 30 to 1000 iterations, the algorithm achieved its maximum number of generations.
  2. The algorithm had discovered a nearly ideal solution when the population approached convergence, meaning that the objective functions improved slightly to none at all over a number of generations.

### 3.3.3 Parallelization Strategy

Parallelization was used to greatly increase the GA's efficiency because of the computationally demanding nature of the multi-objective optimization problem and the large number of evaluations needed for each generation, using multiprocessing to speed up the fitness evaluation process taking use of multi-core platforms, with the main features of the parallelization strategy listed below:

* To divide the workload of fitness evaluations over several processor cores, we put in place a multiprocessing pool, with every member of the population required to have their tilt and azimuth angles evaluated for fitness. We made it possible to execute the method for huge populations and many generations by parallelizing these evaluations, which greatly decreased the amount of time needed to process each generation.
* To guarantee that all processes have access to the required parameters and system configurations, each worker in the multiprocessing pool was initialized with a shared optimization context, being able to assess the objective functions independently and concurrently without doing duplicate computations thanks to this common context, which increased productivity overall.
* Process-based parallelism was used to parallelize the fitness evaluations, which are the most computationally costly step in the optimization process, with the total processing time each generation being decreased by using distinct processes for each fitness evaluation of a population member.
* Finally, the optimization process was significantly sped up by the usage of parallelization. Depending on the number of available cores, the parallelization offered an acceleration factor of four to eight times on multi-core systems, improving the algorithm's scalability by enabling the optimization process to manage bigger populations and more generations in a reasonable amount of time.

## 3.4 Detailed Physical Modeling

The simulation environment featured a comprehensive physical modeling framework for photovoltaic systems to guarantee the precision and realism of the optimization process, taking into consideration meteorological, temporal, and geographical factors to account for the dynamic nature of solar resources, making it possible for the system to accurately synchronize patterns of building consumption and solar energy generation by simulating real-world solar activity with great temporal resolution.

### 3.4.1 Solar Resource and Position Calculation

The accurate determination of the sun position during the simulation period was a crucial part of the physical modeling framework, with the Solar Position Algorithm (SPA), being created by the National Renewable Energy Laboratory (NREL) and accessible through the pvlib Python module, was built in order to accomplish this. In order to calculate the incident solar irradiance on tilted surfaces, the SPA offers extremely precise estimates of the sun's position at any given time and place.

Hourly time resolution was used for the simulation, guaranteeing a balance between temporal granularity and computing performance, and while still reasonable for long-term optimization scenarios, like annual simulations, it proved adequate to capture daily cycles of solar generation. The solar azimuth angle (the compass direction from which the sunlight is coming) and the solar zenith angle (the angle between the sun and the vertical) were two important factors that were computed in this step, being essential for calculating how much direct and diffuse irradiance the panel receives since they establish how the sun is oriented in relation to the Earth's surface.

The following is a functional expression for the mathematical model that underlies solar position calculations:

Solar Position=f(latitude,longitude,timestamp,altitude)

Here, the sun's location is determined by combining timestamps with local elevation and geographic coordinates, being extracted from the site-specific simulation data in order to guarantee that the model accurately represented the environmental conditions of the location under study. We could recreate the angle interaction between sunlight and PV panel surfaces at every hour by precisely modeling the solar position throughout the year, making it possible to calculate the incident solar energy precisely and, consequently, the power output of PV modules in different orientations. The integrity of the ensuing optimization stages depended on this high-fidelity method.

### 3.4.2 Irradiance Modeling

Accurately calculating the solar irradiance incident on the tilted plane of the PV modules, also known as plane-of-array (POA) irradiance, is an important component of physically modeling the performance of photovoltaic systems. In this procedure, horizontal irradiance measurements, which are usually obtained from meteorological datasets, are converted into numbers that represent the tilt and orientation of the installed panels. We used the Hay-Davies model, a popular transposition model that is renowned for striking a compromise between computational ease and precision when breaking down and reprojecting irradiance components, for this purpose.

Direct beam irradiance, diffuse sky irradiance, and ground-reflected irradiance are the three main parts that make up the total POA irradiance, with the cosine of the angle of incidence (AOI) between the sun's rays and the panel surface being used to project the direct normal irradiance (DNI) onto the slanted plane, which yields the direct component. The anisotropy index (Ai) and the sky view factor (SVF), taking into account the uneven distribution of diffuse radiation and the panel's exposure to the sky, are used to alter the diffuse component, which takes scattered sunlight into account. The ground reflectance coefficient (ρ) and a geometrical term that depends on panel tilt regulate the ground-reflected component, which is represented as a fraction of the global horizontal irradiance (GHI).

The following is the full mathematical formula for estimating POA irradiance:

Where:

* POA is the plane-of-array irradiance (W/m²)
* DNI is the direct normal irradiance (W/m²)
* AOI is the angle of incidence (degrees)
* DHI is the diffuse horizontal irradiance (W/m²)
* Rb is the ratio of beam radiation on the tilted surface to the horizontal
* Ai is the anisotropy index
* SVF is the sky view factor
* GHI is the global horizontal irradiance (W/m²)
* ρ is the ground reflectance (typically ranging from 0.2 to 0.6 depending on the surface)
* tilt is the panel’s inclination from horizontal (degrees)

This model, accounting for both isotropic and anisotropic diffuse components and accounts for seasonal and geometric fluctuations, enables an accurate hourly assessment of incoming solar energy on a tilted surface, being essential in laying the groundwork for accurate production projections, which in turn guided the more comprehensive optimization of energy storage architecture and system orientation.

## 3.5 Panel Performance Model

The precise characteristics of the Sharp ND-R240A5 polycrystalline solar panel, a commercially accessible module renowned for its cost-effectiveness, serve as the foundation for the photovoltaic system's performance modeling, having a rated power output of 240 Wp under Standard Test Conditions (STC) and physical dimensions of 1.652 m × 0.994 m, resulting in a surface area of 1.642 m². Its efficiency of 14.6% is adequate for simulating mid-range PV deployments and is typical for polycrystalline technology.

The performance of panels in the actual world is greatly influenced by thermal impacts, with the output power of the panel tends to decrease as the temperature rises. We used a cell temperature model based on the Nominal Operating Cell Temperature (NOCT), which is 47.5°C for this panel, to account for this, using the link between ambient temperature (Tamb) and irradiation levels to determine cell temperature (Tₐ):

Performance metrics can be thermally adjusted in real time thanks to this estimation. The panel's power output model uses the following formula to calculate DC power as a function of temperature effects, surface area, panel efficiency, and irradiance:

Where:

* Pdc​ is the panel’s DC power output,
* Irr is the effective irradiance (W/m²),
* η is the nominal panel efficiency (14.6%),
* A is the panel area (1.642 m²),
* γ is the temperature coefficient of power, equal to −0.0044/°C,
* Tcel is the estimated cell temperature (°C).

A detailed basis for changing voltage-current behavior under varying climatic conditions is provided by the panel's particular temperature coefficients for various electrical properties, such as Pmax at −0.44%/°C, Voc at −0.329%/°C, and Isc at +0.038%/°C.

## 3.6 System Losses Model

Non-idealities and operational losses that lower energy output in actual installations must be taken into consideration when accurately estimating system performance, including a range of loss mechanisms in our simulation system to represent realistic performance expectations.

A 2% energy reduction was employed to estimate the soiling loss brought on by dust and particle deposition on panel surfaces. Based on typical residential and urban site characteristics, shading losses of 3% were predicted, potentially being caused by self-shading in dense arrays or partial obstruction by neighboring objects. In order to depict current imbalances across panels, which are particularly pertinent in diverse operating conditions, an additional 2% mismatch loss was added, with an additional 2% coming from DC wiring losses brought on by resistive heating in the wires, and an estimated 3% coming from reflection losses brought on by incident light reflecting off the glass surface. Unless otherwise noted, these loss categories were combined to create a total system loss factor of roughly 11.5%, being applied consistently throughout simulations.

System-level power calculations were significantly improved using the inverter efficiency model, with efficiency charts from the manufacturer, which change based on the load ratio, being used to calculate inverter losses. Instead of assuming a fixed nominal value, this dynamic modeling approach enables efficiency to change with input power levels. To replicate situations in which the DC input surpasses the inverter's capability and the power output is truncated, clipping losses were also incorporated, and when both electrical and physical loss mechanisms are combined, the simulation of PV performance under various load and environmental circumstances becomes more reliable and precise.

## 3.7 Consumption-Based Weighting System

Creating a temporal weighting mechanism to give priority to aligning production with times of high demand was a significant innovation:

### 3.7.1 Weighting Factor Calculation

def calculate\_weighting\_factors(df):

"""

Calculate the weighting factors based on the building's consumption data.

Weights are assigned only during daylight hours (solar zenith angle < 90°).

"""

# Daylight condition: zenith angle less than 90 degrees

daylight\_mask = df['zenith'] < 90

# Initialize weighting factors to zero

df['weighting\_factor'] = 0.0

# Extract Load values during daylight only

daylight\_load = df.loc[daylight\_mask, 'Load (kW)']

# Normalize load values during daylight hours (0–1 scale)

if not daylight\_load.empty:

min\_load = daylight\_load.min()

max\_load = daylight\_load.max()

if max\_load != min\_load:

normalized\_load = (daylight\_load - min\_load) / (max\_load - min\_load)

else:

normalized\_load = 1 # If consumption is constant during daylight hours

# Assign normalized weights during daylight hours

df.loc[daylight\_mask, 'weighting\_factor'] = normalized\_load

### 3.7.2 Weighted Mismatch Calculation

# Calculate mismatch (production - consumption)

df['mismatch'] = df['ac\_power\_output'] / 1000 - df['Load (kW)']

# Calculate weighted absolute mismatch

df['weighted\_mismatch'] = df['weighting\_factor'] \* np.abs(df['mismatch'])

# Total weighted mismatch

total\_weighted\_mismatch = df['weighted\_mismatch'].sum()

3.8 Pareto Front Analysis and Solution Selection

The multi-objective optimization produces a set of non-dominated solutions (Pareto front):

3.8.1 Pareto Optimality Definition

If no other solution can enhance one goal without making the other worse, then that solution is Pareto-optimal.

### 3.8.2 Balanced Solution Selection Algorithm

def select\_balanced\_solution(pareto\_front):

"""

Selects the solution from the Pareto front that clearly balances both objectives.

"""

mismatch\_vals = np.array([ind.fitness.values[0] for ind in pareto\_front])

production\_vals = np.array([ind.fitness.values[1] for ind in pareto\_front])

# Normalize mismatch and production clearly

mismatch\_norm = (mismatch\_vals - mismatch\_vals.min()) / (mismatch\_vals.ptp() + 1e-9)

production\_norm = (production\_vals - production\_vals.min()) / (production\_vals.ptp() + 1e-9)

# Balance score with emphasis on production

balance\_score = mismatch\_norm \* 0.3 + (1 - production\_norm) \* 0.7

best\_idx = np.argmin(balance\_score)

return pareto\_front[best\_idx]

## 3.9 Seasonal Analysis Methodology

A thorough seasonal analysis approach was created in order to assess the effects of temporal variability on photovoltaic (PV) system performance, making it possible for the study to pinpoint and measure variations in energy production, consumption, and storage requirements during the course of the year. In areas with unique climatic patterns, where solar irradiance and load profiles vary significantly from month to month, seasonal analysis is especially pertinent.

### 3.9.1 Seasonal Categorization

Based on calendar months, the year was split into four meteorological seasons for the purposes of this analysis, with December through February considered winter, March through May as spring, June through August as summer, and September through November as fall, making it possible to evaluate solar performance specifically in response to variations in sun angles, daylight duration, and seasonally specific weather. Because the needs for heating and cooling tend to change with the seasons, the seasonal segmentation likewise matched consumption trends.

### 3.9.2 Seasonal Metrics Calculation

To evaluate how effectively the PV system matched building energy requirements, a number of critical performance measures were computed inside each seasonal segment, providing a high-level perspective of energy balance and covered total seasonal production and consumption, determining two indicators:

* The self-consumption ratio, measuring the proportion of solar energy used locally and not exported to the grid, is computed as follows:
* The self-sufficiency ratio, quantifying the percentage of total energy consumption that is satisfied by solar power, is computed as follows:

In addition to these ratios, the research identified periods of required grid import or possible grid export by capturing seasonal patterns of energy surplus and deficit, creating daily and hourly profiles to investigate the temporal alignment (or misalignment) between solar availability and load demand.

### 3.9.3 Seasonal Battery Requirements

The discrepancy between production and consumption was subsequently addressed by evaluating battery requirements independently for each season, with the required storage capacity determined by calculating daily energy balance profiles using a simulation-based method, aiming to determine how much energy storage was required to compensate for daily deficits while accounting for fluctuations in load and irradiance. The analysis employed a 95th percentile capacity sizing approach to account for typical fluctuation and guarantee robustness against outlier days, preventing needless oversizing while preserving dependability by guaranteeing that the chosen battery size can cover 95% of daily shortage scenarios in each season. Lastly, the charging and discharging power requirements were also estimated in order to determine the dynamic operating range that the battery system would need to support and to ensure compliance with inverter and load characteristics.

## 3.10 Battery Sizing Methodology

A simulation-based iterative battery sizing process was created in order to precisely identify the ideal battery size for optimizing solar self-consumption and reducing grid dependency, with a sophisticated evaluation of storage requirements under actual energy flow scenarios being made possible by the method's integration of performance and economic criteria with extensive operational modeling.

### 3.10.1 Battery Simulation Parameters

A realistic set of technological assumptions that mirrored those of commercial lithium-ion battery systems served as the foundation for the simulation model, with a 90% round-trip efficiency, which took into consideration energy losses throughout the charging and discharging stages, being one of the crucial factors. In accordance with standard manufacturer criteria to guarantee battery longevity and safety, the depth of discharge (DoD) was restricted to 80%. In order to provide a neutral starting point for daily cycles, the simulation started with an initial state of charge (SoC) of 50%, the battery capacity search range covering 1 to 100 kWh, providing flexibility in finding both vast and minimum storage options with a step size of 1 to 2 kWh based on resolution requirements.

### 3.10.2 Energy Flow Simulation Algorithm

To monitor the dynamic interplay between solar generation, building consumption, and storage behavior, a full-year, hourly resolution simulation was run for each potential capacity, with the algorithm proceeding in a methodical manner as follows:

1. The first energy match between production and demand was determined to be direct self-consumption.
2. Any leftover excess energy was marked for possible storage, and any deficiencies showed unfulfilled need.
3. The battery was charged using excess energy, accounting for round-trip efficiency losses.
4. The battery was discharged to make up the difference during hours of deficit, subject to DoD restrictions and the battery's current level of charge.
5. Robust examination of operational performance was made possible by the simulation, which monitored battery cycling behavior, grid import/export volumes, and hourly state of charge.

### 3.10.3 Metrics for Optimization

A number of critical performance parameters were calculated to assess each battery capacity's efficacy, including increases in self-sufficiency and self-consumption, quantifying the effect of storage on energy independence. In order to assist life cycle cost analysis, battery wear across the simulation period was estimated using equivalent full cycles, with the utilization factor, which shows how effectively each battery size was being used, being computed as the ratio of used capacity to total installed capacity. Economic metrics like ROI and payback duration were also calculated, providing vital information for feasibility and cost-benefit analyses.

### 3.10.4 Optimal Capacity Selection Criteria

A weighted scoring approach that struck a compromise between technical performance and economic viability was used to determine the ideal battery capacity, with a solution requiring to have a significant effect on energy independence and reach a minimum self-sufficiency level of 50% in order to be eligible, and the ideal capacity among all qualifying options being determined to be the one with the lowest payback period. Even though it meant a longer return period, the model automatically switched to the size that offered the highest possible level of self-sufficiency in situations where no battery configuration satisfied the criterion, making sure that system efficacy and investment value were pragmatically balanced.

### 3.11 Economic and Efficiency Analysis

The methodological framework's last element comprised a thorough evaluation of the PV-battery system's performance and economics, aiming to assess the technical effectiveness and resource usage of several system configurations in addition to their financial feasibility, and provide a fair and thorough assessment of suggested solutions by combining engineering and economic variables.

### 3.11.1 Initial Investment Calculation

The main cost components related to PV and battery deployment were combined to estimate the initial capital expenditure, with the cost of each panel fixed at $250, which is the going rate for premium polycrystalline modules, while the cost of installation materials and labor, including mounting structures, wiring, and labor, was $150 per panel. Including both string inverters and inverter-based control systems, the estimated cost of the inverter was $120 per kW of installed PV capacity, with the cost of battery storage, including energy storage modules and related balance of plant, estimated to be $500 per kWh. It was expected that each panel would cost $50 for additional balance of system (BoS) expenses, such as switches, connectors, and monitoring systems.

### 3.11.2 Annual Cash Flow Modeling

An yearly cash flow model was created in order to evaluate the investment's long-term financial success, with electricity cost savings being computed by substituting grid-supplied electricity with direct self-consumption of solar energy. Depending on the regulatory setting, feed-in tariffs or net metering values were used to incorporate additional grid export profits, and according to industry standards for residential and commercial PV systems, the expected yearly maintenance expenses were 0.5% of the initial investment. In order to account for aging panels, the model additionally included performance degradation, assuming an annual efficiency loss of 0.5%. A typical increasing trend in energy markets was forecast for power price hikes of 3% annually and inflation of 2% annually, enabling the precise calculation of the long-term advantages and hazards of the investment.

### 3.11.3 Financial Metrics

The attractiveness of the investment was measured using important financial parameters. One of the main indicators of economic viability is net present value (NPV), calculating the total discounted worth of future cash flows. The effective interest rate received on the investment over its lifetime was represented by the Internal Rate of Return (IRR), with a straightforward percentage indicator of profitability being the return on investment (ROI), and the payback period showed how many years it would take for savings and income to recoup the initial outlay. In order to enable direct comparison with utility rates and other energy sources, the Levelized Cost of Electricity (LCOE), representing the average cost per kilowatt-hour of energy generated over the system's lifetime, was also calculated.

### 3.11.4 Efficiency Metrics

System performance was assessed using a set of technical efficiency criteria in addition to financial analysis, with system efficiency taking into consideration all loss factors from irradiation to usable AC output, whereas panel efficiency was defined as the ratio of electrical output to solar input under established test settings, while taking into account real-world circumstances, the performance ratio (PR) represented the actual production in relation to the theoretical maximum. While specific yield (kWh/m2) offered a normalized output per surface area, facilitating comparisons across panel types and layouts, system yield (kWh/kWp) quantified the overall yearly energy output per unit of installed capacity. In spatially limited situations, the capacity factor, being the ratio of actual to prospective generation over time, and land usage efficiency (kWh/m²/year) were especially important, providing when combined a comprehensive picture of spatial use and energy conversion efficiency.

## 3.12 Integrated Results and Interpretation

Combining all of the simulation outputs, optimization outcomes, and performance assessments into a logical interpretative framework was the methodological framework's last stage, making possible a comprehensive decision-making based on both technical and economic viewpoints. First, the Pareto front produced by the NSGA-II algorithm was used to determine the ideal panel design, in particular the tilt and azimuth angles, reflecting trade-offs between minimizing energy mismatch and optimizing overall production, being chosen because they were close to optimal self-consumption patterns and energy generation targets, guaranteeing that with few post-installation modifications, the chosen system orientation would offer robust year-round performance.

Second, temporal variability in energy production and consumption was highlighted using seasonal performance measurements, with the seasonal analysis, taking into consideration shifting solar angles, weather patterns, and load dynamics, showing how system performance varied throughout the year. Strategic choices like scheduling high-load tasks during high-generation seasons and adaptive energy management were influenced by these insights. Third, the research yielded suggestions for battery sizing that were based on economic feasibility as well as energy flow models, and in order to balance capital investment and lifespan utilization, the sizing was determined by reaching a goal self-sufficiency level, with system designers being able to make well-informed trade-offs since each suggested capacity level was backed by measures like economic return parameters, equivalent full cycles, and utilization factor.

Fourth, a thorough economic analysis was carried out, taking into account all pertinent financial metrics such as Levelized Cost of Electricity (LCOE), Payback Period, and Return on Investment (ROI), with stakeholders being able to measure the investment's profitability using these measures in real-world scenarios, such as deterioration, inflation, and rising energy prices, making sure that configurations that were technically ideal also satisfied the needs of economic sustainability. In order to assess technical performance, the framework provided a set of efficiency criteria, with a thorough grasp of how well the system transformed solar resources into useful energy being made possible by metrics including system efficiency, performance ratio, specific yield, and capacity factor, aiding technical benchmarking and ongoing performance monitoring.

## 3.13 Energy Consumption Forecasting

One of the most important tasks in managing contemporary smart networks is accurately estimating energy use. Utility companies can balance loads, optimize generation schedules, lower operating costs, and more successfully integrate renewable energy sources by forecasting future energy demand, with machine learning models emerging as a key instrument in expanding predictive skills beyond conventional statistical techniques, thanks to the growing availability of detailed time-series energy data from smart meters and Internet of Things sensors. In this study, supervised machine learning models built with TensorFlow, a popular open-source machine learning library, and pandas, a Python-based data manipulation toolkit, are used to forecast energy use, using pandas for pre-processing time-series data, including data normalization, resampling, and feature engineering (including calendar variables, lag features, and rolling averages), guaranteeing that the TensorFlow models receive clean, organized, and instructive input data.

For prediction, several machine learning architectures are investigated, using recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM) networks to model sequential relationships in temporal energy data, representing long-term trends like seasonal variations in demand or recurring usage patterns. In order to capture nonlinear interactions between input features, Dense Feedforward Neural Networks (DNNs) are also used as baselines. In increasingly intricate configurations, hybrid models that combine LSTMs and convolutional layers are explored to simultaneously extract long-term trends and short-term patterns, using time-aware cross-validation techniques to split historical energy consumption datasets for training and validation of these models, thus ensuring that training always comes before testing in order to replicate real-world deployment settings. TensorFlow model training uses batch normalization, GPU acceleration, and adaptive optimizers (like Adam) to effectively converge on the best predicting performance, while using standard quantitative measurements designed for regression and time-series contexts to assess forecasting accuracy:

* The mean absolute error (MAE), which is directly interpreted in the same units as energy consumption (e.g., kWh), measures the average magnitude of forecast errors.
* The root The Mean Squared Error (RMSE) is sensitive to outliers in consumption spikes because it assigns greater weight to larger mistakes.
* The Mean Absolute Percentage Error (MAPE), which quantifies mistakes in relative percentage terms, is helpful for comparing model performance across datasets with varying average energy levels.
* When working with consumption values that are close to zero, the Symmetric Mean Absolute Percentage Error (sMAPE) overcomes the drawbacks of MAPE.
* As a general indicator of model fit, the coefficient of determination (R2 Score) shows what percentage of the variation in actual energy use can be explained by the model.

# CHAPTER 4 Application

## 4.1 Code Organization & Structure

A single main script, pv\_simulation.py, serves as the foundation for the modular, maintainable Python application that implements the solar PV optimization framework, being the system's core coordinating several computational phases in a logical pipeline. To encourage clarity and reusability, each simulation step is represented by a functionally separate code block or module.

Constants and imports set system-wide parameters and import required libraries at the start of the structure, going on to include parallel computing-supporting modules for Worker Management and Optimization Context. While the Sun Modeling Functions calculate sun locations and irradiance, the Data Processing Functions deal with input preparation, with PV production estimated via energy calculation functions, which then input the findings into the genetic algorithm-driven Multi-Objective Optimization section. Analysis is carried out in later sections, which also include simulations of battery performance, visualizations, and a final economic evaluation, with all of the modules being connected by the Main Execution Flow, simplifying the entire process from raw inputs to optimization outputs and plotted outcomes. Data moves through a series of clearly defined steps in this pipeline-style architecture, with each stage building on the results of the one before it In order to facilitate smooth data transformation and optimization also making extension easier, allowing for the future integration of new models or datasets.

pv\_simulation.py

├── Constants and Imports

├── Optimization Context and Worker Management

├── Data Processing Functions

├── Solar Modeling Functions

├── Energy Calculation Functions

├── Multi-Objective Optimization

├── Analysis Functions

├── Visualization Functions

├── Battery Modeling Functions

├── Economic Analysis Functions

└── Main Execution Flow

## 4.2 Key Modules & Their Purpose

### 4.2.1 Imports and Dependencies

A strong collection of Python libraries designed for scientific computing, solar modeling, optimization, and system-level administration are the foundation of the application, with some noteworthy imports being:

* Scientific Computing, with the numerical foundation of the program being made up of numpy and scipy, allowing for mathematical modeling and array operations.
* Data Handling, with Pandas being widely used to handle time series data, including load profiles, hourly irradiance, and simulation results.
* Sun Modeling, with pvlib offering trustworthy models For the computation of sun position, irradiance components, and plane-of-array modifications.
* Optimization, with the Distributed Evolutionary Algorithms in Python (DEAP) library being used to implement the NSGA-II algorithm, while scipy.optimize.differential\_evolution is available for alternative search techniques.
* Visualization, with matplotlib and Seaborn being used to create output graphs and figures that facilitate stakeholder reporting as well as technical examination.
* System Integration, with file management, command-line interface, and application flexibility made possible by modules like argparse, os, and sys.
* Parallel Processing, with the multiprocessing module speeding up computationally demanding operations, especially while the genetic algorithm is evaluating fitness.

To ensure consistency and readability across modules, internal constants and parameters are defined in a separate file (constants.py) and imported as needed in addition to external libraries.

Using contextual and historical data, the Energy Forecasting Module is in charge of forecasting both short- and long-term electricity use, making use of TensorFlow-created machine learning models, with Pandas handling feature engineering and preprocessing to guarantee clean, high-quality inputs. By supporting demand forecasts for both residential and commercial properties, the module empowers utilities to take proactive measures in infrastructure planning, load distribution, and peak shaving. Metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are used to continuously assess forecasting accuracy and make sure the models continue to adjust to changes in consumption patterns over time.

### 4.2.2 Configuration & Constants

The method uses both random.seed() and np.random.seed() to set a fixed random seed during initialization in order to guarantee reproducibility of simulation and optimization outcomes, with the application making use of the parameter RANDOM\_SEED = 42. In order to centralize setting and minimize code duplication, important parameters like the Time Interval (in hours), Nominal Operating Cell Temperature (NOCT), and Total Loss Factor are specified and imported from constants.py. The TOTAL\_LOSS\_FACTOR is utilized in DC to AC power conversions, NOCT is essential for cell temperature modeling, and TIME\_INTERVAL\_HOURS establishes the simulation's temporal resolution, bringing all time-based calculations into alignment, with the code providing maintainability and transparency by combining these values.

## 4.3 Core Data Processing Functions

### 4.3.1 Data Loading & Preprocessing

Time-series data can be effectively loaded and preprocessed for use in the solar PV optimization model with the help of the load\_and\_preprocess\_data function, starting by reading the input data from a CSV file and converting the hour column to a datetime object, with the timestamps then starting on January 1st, 2023 with the function then making sure that time zones are handled correctly by localizing the data to the 'Europe/Athens' timezone and using the'shift\_forward' method to account for daylight saving time (DST) transitions, and finally setting the newly formed datetime column as the index.

The method resolves duplicates by retaining just the first occurrence, eliminates rows with missing timestamps (NaT), and imposes an hourly periodicity for the dataset in order to guarantee clean and consistent data, guaranteeing that there are no gaps in the time-series data, by using forward and backward filling to manage missing values in important columns like solar radiation, air temperature, wind speed, and load. The data will be prepared for precise and trustworthy analysis in later phases of the optimization process thanks to this thorough pretreatment procedure.

def load\_config(config\_file):

"""

Load configuration from a YAML file.

Parameters:

- config\_file (str): Path to the YAML configuration file.

Returns:

- config (dict): Configuration parameters.

"""

try:

with open(config\_file, 'r') as file:

config = yaml.safe\_load(file)

logging.info(f"Configuration loaded from {config\_file}")

except Exception as e:

logging.error(f"Error loading configuration file: {e}", exc\_info=True)

raise

return config

def load\_and\_preprocess\_pvgis\_data(pvgis\_file, load\_file):

"""

CRITICAL FIX: Handle multi-year PVGIS data correctly by creating a synthetic year.

"""

logging.info("=== LOADING MULTI-YEAR PVGIS DATA ===")

try:

# Load data

df\_pvgis = pd.read\_csv(pvgis\_file)

logging.info(f"PVGIS data loaded: {len(df\_pvgis)} records")

logging.info(f"PVGIS columns: {list(df\_pvgis.columns)}")

# Load consumption data

df\_load = pd.read\_csv(load\_file)

logging.info(f"Load data loaded: {len(df\_load)} records")

def parse\_pvgis\_time\_FIXED(time\_str):

"""Enhanced time parsing"""

try:

if ':' in time\_str:

date\_part = time\_str.split(':')[0]

time\_part = time\_str.split(':')[1]

else:

date\_part = time\_str[:8]

time\_part = time\_str[8:] if len(time\_str) > 8 else "0000"

year = int(date\_part[:4])

month = int(date\_part[4:6])

day = int(date\_part[6:8])

hour = int(time\_part[:2])

minute = int(time\_part[2:4]) if len(time\_part) >= 4 else 0

return datetime(year, month, day, hour, minute)

except Exception as e:

logging.error(f"Error parsing time '{time\_str}': {e}")

return None

# Parse original times

df\_pvgis['original\_datetime'] = df\_pvgis['time\_UTC'].apply(parse\_pvgis\_time\_FIXED)

df\_pvgis = df\_pvgis.dropna(subset=['original\_datetime'])

logging.info("Creating synthetic year timeline for TMY data with 30-min timestamp correction...")

# Create a standard year timeline (2020 as reference, non-leap year for consistency)

base\_year = 2020

start\_date = datetime(base\_year, 1, 1, 0, 0)

synthetic\_timeline = pd.date\_range(

start=start\_date,

periods=8760, # Exactly one year of hours

freq='h',

tz='UTC'

)

# Map original data to synthetic timeline by month-day-hour

df\_pvgis['month'] = df\_pvgis['original\_datetime'].dt.month

df\_pvgis['day'] = df\_pvgis['original\_datetime'].dt.day

df\_pvgis['hour'] = df\_pvgis['original\_datetime'].dt.hour

df\_pvgis['synthetic\_datetime'] = df\_pvgis.apply(

lambda row: datetime(base\_year, row['month'], row['day'], row['hour'], 0) +

timedelta(minutes=30), # PVGIS timestamps represent middle of hour

axis=1

)

# Sort by synthetic datetime to ensure proper chronological order

df\_pvgis = df\_pvgis.sort\_values('synthetic\_datetime').reset\_index(drop=True)

# CRITICAL: Localize to UTC first, then convert to Athens

df\_pvgis['datetime\_utc'] = pd.to\_datetime(df\_pvgis['synthetic\_datetime']).dt.tz\_localize('UTC')

df\_pvgis['datetime\_athens'] = df\_pvgis['datetime\_utc'].dt.tz\_convert('Europe/Athens')

# VALIDATION: Check timeline consistency

if len(df\_pvgis) != 8760:

logging.error(f"Expected 8760 hours, got {len(df\_pvgis)}")

# Check for duplicates

duplicate\_times = df\_pvgis['synthetic\_datetime'].duplicated().sum()

if duplicate\_times > 0:

logging.warning(f"Found {duplicate\_times} duplicate timestamps - keeping first occurrence")

df\_pvgis = df\_pvgis.drop\_duplicates(subset=['synthetic\_datetime'], keep='first')

# Check for missing hours

expected\_hours = set(range(8760))

actual\_hours = set((df\_pvgis['synthetic\_datetime'] - datetime(base\_year, 1, 1)).dt.total\_seconds() // 3600)

missing\_hours = expected\_hours - actual\_hours

if missing\_hours:

logging.warning(f"Missing {len(missing\_hours)} hours in the dataset")

df\_mapped = pd.DataFrame()

df\_mapped['datetime'] = df\_pvgis['datetime\_athens']

df\_mapped['Air Temp'] = df\_pvgis['T2m']

df\_mapped['SolRad\_Hor'] = df\_pvgis['G(h)'] # Global Horizontal Irradiance

df\_mapped['SolRad\_Dif'] = df\_pvgis['Gd(h)'] # Diffuse Horizontal Irradiance

df\_mapped['WS\_10m'] = df\_pvgis['WS10m']

if 'Gb(n)' in df\_pvgis.columns:

df\_mapped['DNI\_pvgis'] = df\_pvgis['Gb(n)']

logging.info("Using PVGIS-provided DNI data")

# Set datetime as index

df\_mapped.set\_index('datetime', inplace=True)

df\_mapped = df\_mapped.sort\_index()

# CRITICAL FIX: Validate solar data quality

max\_ghi = df\_mapped['SolRad\_Hor'].max()

avg\_ghi = df\_mapped['SolRad\_Hor'].mean()

annual\_ghi = df\_mapped['SolRad\_Hor'].sum() / 1000

logging.info(f"FIXED PVGIS solar data validation:")

logging.info(f" Max GHI: {max\_ghi:.0f} W/m²")

logging.info(f" Average GHI: {avg\_ghi:.0f} W/m²")

logging.info(f" Annual GHI: {annual\_ghi:.0f} kWh/m²")

# Handle load data alignment

if len(df\_load) == len(df\_mapped):

df\_mapped['Load (kW)'] = df\_load['Load (kW)'].values

else:

logging.warning(f"Length mismatch: PVGIS={len(df\_mapped)}, Load={len(df\_load)}")

if 'Load (kW)' in df\_load.columns:

# Create matching timeline for load data

load\_timeline = pd.date\_range(

start=df\_mapped.index[0],

periods=len(df\_load),

freq='h'

)

df\_load\_indexed = df\_load.set\_index(load\_timeline)

load\_reindexed = df\_load\_indexed['Load (kW)'].reindex(df\_mapped.index, method='nearest')

df\_mapped['Load (kW)'] = load\_reindexed.fillna(df\_load['Load (kW)'].mean())

logging.info("Load data aligned using nearest neighbor interpolation")

else:

df\_mapped['Load (kW)'] = 100.0

logging.warning("Load column not found, using default 100 kW")

# Ensure all columns are numeric

numeric\_columns = ['Air Temp', 'SolRad\_Hor', 'SolRad\_Dif', 'WS\_10m', 'Load (kW)']

for col in numeric\_columns:

if col in df\_mapped.columns:

df\_mapped[col] = pd.to\_numeric(df\_mapped[col], errors='coerce')

if df\_mapped[col].isnull().any():

if col in ['SolRad\_Hor', 'SolRad\_Dif']:

df\_mapped[col] = df\_mapped[col].fillna(0)

else:

df\_mapped[col] = df\_mapped[col].interpolate().bfill()

# Add E\_ac column

if 'E\_ac' not in df\_mapped.columns:

df\_mapped['E\_ac'] = 0.0

time\_diffs = df\_mapped.index.to\_series().diff().dropna()

time\_diffs\_hours = time\_diffs.dt.total\_seconds() / 3600

logging.info(f"=== FIXED TIME INTERVAL VALIDATION ===")

logging.info(f"Time interval statistics after fix:")

logging.info(f" Mean: {time\_diffs\_hours.mean():.3f} hours")

logging.info(f" Std: {time\_diffs\_hours.std():.3f} hours")

logging.info(f" Min: {time\_diffs\_hours.min():.3f} hours")

logging.info(f" Max: {time\_diffs\_hours.max():.3f} hours")

non\_hourly = time\_diffs\_hours[(time\_diffs\_hours < 0.95) | (time\_diffs\_hours > 1.05)]

if len(non\_hourly) > 0:

logging.warning(f"Still found {len(non\_hourly)} non-hourly intervals after fix!")

else:

logging.info("✓ All time intervals are now exactly 1 hour")

# Final validation

logging.info(f"=== FIXED PVGIS DATA SUMMARY ===")

logging.info(f"Records: {len(df\_mapped)}")

logging.info(f"Time range: {df\_mapped.index.min()} to {df\_mapped.index.max()}")

logging.info(f"Timezone: {df\_mapped.index.tz}")

# Check if solar noon occurs around 12-13 local time

summer\_data = df\_mapped[df\_mapped.index.month == 6]

if len(summer\_data) > 0:

peak\_solar\_hour = summer\_data.groupby(summer\_data.index.hour)['SolRad\_Hor'].mean().idxmax()

logging.info(f"Peak solar occurs at hour: {peak\_solar\_hour} (should be 12-13 for Athens)")

return df\_mapped

except Exception as e:

logging.error(f"Fixed PVGIS processing failed: {e}", exc\_info=True)

raise

### 4.3.2 Solar Position Calculation

In order to precisely simulate solar irradiance on slanted surfaces, the solar zenith and azimuth angles must be calculated using the calculate\_solar\_position function, with the function calling get\_solarposition, implementing the NREL Solar Position Algorithm (SPA), with the help of the pvlib package. With an accuracy of ±0.0003° for sun position estimations between 2000 and 2100, this technique is incredibly accurate, with the function accepting as input a dataframe containing the latitude and longitude of the place together with time-indexed solar data. The dataframe is output with two extra columns: azimuth (the angle of the sun's direction with respect to the north) and zenith (the solar zenith angle), and since the angle of incidence affects energy output, these solar position numbers are very important for figuring out the incident solar radiation on the panel surface.

def calculate\_solar\_position(df, latitude, longitude):

"""

FIXED: Calculate solar position with proper validation for Athens.

"""

logging.info("=== FIXED SOLAR POSITION CALCULATION ===")

try:

if df.index.tz is None:

logging.error("DataFrame must have timezone-aware index!")

raise ValueError("DateTime index must have timezone information")

logging.info(f"Input data timeline: {df.index.min()} to {df.index.max()}")

logging.info(f"Data timezone: {df.index.tz}")

# Calculate solar position

solar\_position = pvlib.solarposition.get\_solarposition(

df.index, latitude, longitude, method='nrel\_numpy'

)

df['zenith'] = solar\_position['apparent\_zenith']

df['azimuth'] = solar\_position['azimuth']

logging.info("=== FIXED SOLAR POSITION VALIDATION ===")

# Check solar noon timing

daylight\_mask = (df['zenith'] < 85) & (df['SolRad\_Hor'] > 50)

if daylight\_mask.any():

df\_daylight = df[daylight\_mask].copy()

df\_daylight['hour\_decimal'] = (df\_daylight.index.hour +

df\_daylight.index.minute/60.0)

# Find solar noon for each day

daily\_solar\_noon = []

for date in pd.date\_range(df.index.min().date(), df.index.max().date(), freq='D'):

day\_data = df\_daylight[df\_daylight.index.date == date.date()]

if len(day\_data) > 0:

min\_zenith\_idx = day\_data['zenith'].idxmin()

noon\_hour = day\_data.loc[min\_zenith\_idx, 'hour\_decimal']

daily\_solar\_noon.append(noon\_hour)

if daily\_solar\_noon:

avg\_solar\_noon = sum(daily\_solar\_noon) / len(daily\_solar\_noon)

logging.info(f"Average solar noon: {avg\_solar\_noon:.2f}h")

if 11.5 <= avg\_solar\_noon <= 13.5: # Allow for DST and equation of time

logging.info("✓ Solar noon timing is reasonable for Athens")

solar\_noon\_ok = True

else:

logging.warning(f"⚠ Solar noon at {avg\_solar\_noon:.2f}h is unusual for Athens!")

solar\_noon\_ok = False

else:

solar\_noon\_ok = True

else:

solar\_noon\_ok = True

# CRITICAL FIX: Orientation validation with proper DNI calculation

logging.info("=== FIXED ORIENTATION VALIDATION TEST ===")

try:

# Calculate DNI if not present

if 'DNI' not in df.columns:

if 'DNI\_pvgis' in df.columns:

df['DNI'] = df['DNI\_pvgis']

logging.info("Using PVGIS DNI for validation")

else:

dni = pvlib.irradiance.disc(

ghi=df['SolRad\_Hor'],

solar\_zenith=df['zenith'],

datetime\_or\_doy=df.index

)['dni']

df['DNI'] = dni

logging.info("Calculated DNI using disc model for validation")

# Get DNI extra for irradiance calculations

dni\_extra = pvlib.irradiance.get\_extra\_radiation(df.index, method='nrel')

# FIXED: Test orientations that should show south is optimal

test\_orientations = [

(30, 90, "East"), # Should be lower

(30, 135, "Southeast"), # Should be lower than south

(30, 180, "South"), # Should be HIGHEST for max energy

(30, 225, "Southwest"), # Should be lower than south

(30, 270, "West") # Should be lower

]

irradiation\_results = {}

for tilt, azimuth, name in test\_orientations:

try:

irradiance\_data = pvlib.irradiance.get\_total\_irradiance(

surface\_tilt=tilt,

surface\_azimuth=azimuth,

solar\_zenith=df['zenith'],

solar\_azimuth=df['azimuth'],

dni=df['DNI'],

ghi=df['SolRad\_Hor'],

dhi=df['SolRad\_Dif'],

dni\_extra=dni\_extra,

model='haydavies'

)

annual\_irradiation = irradiance\_data['poa\_global'].sum() / 1000

irradiation\_results[name] = annual\_irradiation

logging.info(f" {name} ({azimuth}°): {annual\_irradiation:,.0f} kWh/m²")

except Exception as e:

logging.error(f"Error calculating irradiance for {name}: {e}")

irradiation\_results[name] = 0

# CRITICAL: Check if South produces maximum irradiation

if len(irradiation\_results) > 0:

max\_orientation = max(irradiation\_results.keys(),

key=lambda k: irradiation\_results[k])

south\_value = irradiation\_results.get('South', 0)

max\_value = max(irradiation\_results.values())

logging.info(f"Maximum irradiation orientation: {max\_orientation}")

logging.info(f"South irradiation: {south\_value:,.0f} kWh/m²")

logging.info(f"Maximum irradiation: {max\_value:,.0f} kWh/m²")

if max\_orientation == 'South':

logging.info("✓ VALIDATION PASSED: South produces maximum irradiation")

orientation\_ok = True

else:

# Check how much better the max is than south

if south\_value > 0:

excess = ((max\_value - south\_value) / south\_value) \* 100

logging.error(f"✗ VALIDATION FAILED: {max\_orientation} produces {excess:.1f}% more than South!")

logging.error("This indicates a problem with:")

logging.error(" - Solar position calculations")

logging.error(" - Timezone conversion")

logging.error(" - Data quality")

orientation\_ok = False

else:

logging.error("✗ VALIDATION FAILED: South produces zero irradiation!")

orientation\_ok = False

else:

orientation\_ok = False

except Exception as e:

logging.error(f"Orientation validation failed: {e}")

orientation\_ok = False

# Overall validation result

validation\_passed = solar\_noon\_ok and orientation\_ok

if validation\_passed:

logging.info("✓ OVERALL SOLAR POSITION VALIDATION: PASSED")

else:

logging.error("✗ OVERALL SOLAR POSITION VALIDATION: FAILED")

logging.error("Solar calculations may be incorrect - optimization results will be wrong!")

return df, validation\_passed

except Exception as e:

logging.error(f"Solar position calculation failed: {e}", exc\_info=True)

### 4.3.3 DNI Calculation

One important aspect of solar irradiance that is perpendicular to the solar panel is the Direct Normal Irradiance (DNI), which is estimated by the calculate\_dni function, with the DNI being obtained from solar zenith angles and Global Horizontal Irradiance (GHI) using the pvlib function disc(). The DISC (Direct Insolation Simulation Code) model is appropriate for predicting DNI from easily accessible GHI measurements as it uses the relationship between the clearness index and the ratio of DNI to GHI. Each time step's DNI values are calculated by the function and appended to the dataframe as the DNI column. Accurately calculating the panel's energy output based on both direct and diffuse solar contributions requires this computation.

def calculate\_dni(df):

"""

Calculate DNI using disc model.

Parameters:

- df (DataFrame): DataFrame with necessary irradiance and solar position data.

Returns:

- df (DataFrame): DataFrame with 'DNI' column added.

"""

required\_columns = ['SolRad\_Hor', 'zenith']

for col in required\_columns:

if col not in df.columns:

logging.error(f"'{col}' column is missing in the DataFrame.")

raise ValueError(f"'{col}' column is missing in the DataFrame.")

try:

# Check if PVGIS already provided DNI data

if 'DNI\_pvgis' in df.columns:

logging.info("Using PVGIS-provided DNI data")

df['DNI'] = df['DNI\_pvgis']

# Validate PVGIS DNI data

max\_dni = df['DNI'].max()

mean\_dni = df['DNI'].mean()

logging.info(f"PVGIS DNI validation:")

logging.info(f" Max DNI: {max\_dni:.0f} W/m²")

logging.info(f" Mean DNI: {mean\_dni:.0f} W/m²")

# Check for reasonable DNI values

if max\_dni > 1200:

logging.warning(f"Very high DNI values detected: max = {max\_dni:.0f} W/m²")

if max\_dni < 800:

logging.warning(f"Low DNI peak detected: max = {max\_dni:.0f} W/m²")

# Handle negative or missing values

negative\_count = (df['DNI'] < 0).sum()

if negative\_count > 0:

logging.warning(f"Found {negative\_count} negative DNI values - setting to 0")

df['DNI'] = df['DNI'].clip(lower=0)

nan\_count = df['DNI'].isnull().sum()

if nan\_count > 0:

logging.warning(f"Found {nan\_count} missing DNI values - filling with 0")

df['DNI'] = df['DNI'].fillna(0)

else:

# Calculate DNI using pvlib's disc() function

logging.info("Calculating DNI using disc model")

dni = pvlib.irradiance.disc(

ghi=df['SolRad\_Hor'],

solar\_zenith=df['zenith'],

datetime\_or\_doy=df.index

)['dni']

df['DNI'] = dni

# Validate DNI timing regardless of source

daylight\_mask = df['zenith'] < 85

if daylight\_mask.any():

df\_test = df[daylight\_mask].copy()

df\_test['hour'] = df\_test.index.hour

hourly\_dni = df\_test.groupby('hour')['DNI'].mean()

hourly\_ghi = df\_test.groupby('hour')['SolRad\_Hor'].mean()

dni\_peak\_hour = hourly\_dni.idxmax()

ghi\_peak\_hour = hourly\_ghi.idxmax()

logging.info(f"DNI peak at: {dni\_peak\_hour}h ({hourly\_dni.max():.0f} W/m²)")

logging.info(f"GHI peak at: {ghi\_peak\_hour}h ({hourly\_ghi.max():.0f} W/m²)")

if dni\_peak\_hour < 10 or dni\_peak\_hour > 15:

logging.warning(f"DNI peak at {dni\_peak\_hour}h seems unusual for Athens")

logging.warning(f"This suggests potential datetime conversion issues")

else:

logging.info(f"DNI peak timing looks reasonable")

logging.info("DNI processing completed and added to DataFrame.")

# Final DNI statistics

annual\_dni = df['DNI'].sum() / 1000 # kWh/m²

logging.info(f"Annual DNI: {annual\_dni:.0f} kWh/m² (typical range for Athens: 1800-2200)")

except Exception as e:

logging.error(f"Error calculating/processing DNI: {e}", exc\_info=True)

raise

return df

## 4.4. Solar Energy Modeling Functions

### 4.4.1 Total Irradiance Calculation

Taking into account both direct and diffuse radiation, the calculate\_total\_irradiance function determines the total irradiance incident on a solar panel, by utilizing the Hay-Davies model, a popular technique for determining plane-of-array (POA) irradiance to achieve this, taking as inputs the global horizontal irradiance (GHI), diffuse horizontal irradiance (DHI), direct normal irradiance (DNI), and azimuth and zenith angles of the sun, furthermore taking into account extraterrestrial radiation (dni\_extra), which is essential for precise modeling of solar irradiance.The function outputs the total irradiance on the panel surface using the Hay-Davies model, and this value is saved in the dataframe's total\_irradiance column, with the direct beam, sky diffuse, and ground-reflected radiation all contributing to this total irradiance, and each being essential for calculating the system's solar energy production.

def calculate\_total\_irradiance(df, tilt\_angle, azimuth\_angle, dni\_extra):

"""

FIXED: Calculate total irradiance with azimuth correction for Athens.

"""

required\_columns = ['zenith', 'azimuth', 'DNI', 'SolRad\_Hor', 'SolRad\_Dif']

for col in required\_columns:

if col not in df.columns:

logging.error(f"'{col}' column is missing in the DataFrame.")

raise ValueError(f"'{col}' column is missing in the DataFrame.")

try:

# CRITICAL FIX: Apply azimuth correction for Athens solar position data

# The solar azimuth data appears to be offset by ~20°

# Apply -15° correction to surface azimuth to compensate

corrected\_azimuth = azimuth\_angle - 15.0

# Ensure azimuth stays in 0-360° range

if corrected\_azimuth < 0:

corrected\_azimuth += 360

elif corrected\_azimuth >= 360:

corrected\_azimuth -= 360

logging.info(f"AZIMUTH FIX: {azimuth\_angle}° → {corrected\_azimuth}° (applied -15° correction)")

irradiance\_data = pvlib.irradiance.get\_total\_irradiance(

surface\_tilt=tilt\_angle,

surface\_azimuth=corrected\_azimuth, # Use corrected azimuth

solar\_zenith=df['zenith'],

solar\_azimuth=df['azimuth'],

dni=df['DNI'],

ghi=df['SolRad\_Hor'],

dhi=df['SolRad\_Dif'],

dni\_extra=dni\_extra,

model='haydavies'

)

df['total\_irradiance'] = irradiance\_data['poa\_global']

logging.info(f"Total irradiance calculated with CORRECTED tilt {tilt\_angle}° and azimuth {corrected\_azimuth}° (originally {azimuth\_angle}°).")

except Exception as e:

logging.error(f"Error calculating total irradiance: {e}", exc\_info=True)

raise

return df

### 4.4.2 Energy Production Calculation

The Sharp ND-R240A5 solar panels' energy conversion process from irradiance to electrical energy production is modeled by the calculate\_energy\_production function, starting by applying the panels' area, efficiency, and rated power, among other physical factors, and after adjusting for system losses such soiling and shading, it first determines the effective irradiance. The Nominal Operating Cell Temperature (NOCT) approach, which considers the irradiance level and the ambient air temperature, is then used to determine the temperature of the solar cells, with the function then using a temperature coefficient for power output to modify each panel's DC power output according to the impact of temperature on panel efficiency.

Each panel's DC power output is determined, and the system's overall power output is scaled with the number of panels, with the efficiency of the inverter being then taken into account when converting this value into AC power, and a clipping mechanism makes sure the output doesn't go beyond the maximum capacity of the inverter. The function monitors a number of energy measures, including DC energy (Edc), AC energy (Eac), incident energy (Eincident), and effective energy (Eeffective), while also analyzing the system's performance ratio (PR), which gauges how well the system converts energy, and calculates losses at each stage, including those caused by shade and soiling, temperature impacts, and inverter inefficiencies.

def calculate\_energy\_production(df, number\_of\_panels, inverter\_params):

"""

IMPROVED energy production calculation with realistic losses for Athens.

"""

required\_columns = ['total\_irradiance', 'Air Temp']

for col in required\_columns:

if col not in df.columns:

logging.error(f"'{col}' column is missing in the DataFrame.")

raise ValueError(f"'{col}' column is missing in the DataFrame.")

try:

# Panel parameters (Sharp ND-R240A5)

panel\_area = 1.642 # m²

panel\_efficiency\_stc = 0.146 # 14.6% at STC

panel\_power\_stc = 240 # Wp at STC

TEMP\_COEFF\_PMAX = -0.0044 # -0.44%/°C

NOCT = 47.5 # °C

# System parameters

total\_panel\_area = panel\_area \* number\_of\_panels

total\_system\_power\_stc = panel\_power\_stc \* number\_of\_panels

# REALISTIC loss factors for Athens (Mediterranean climate)

soiling\_factor = 0.96 # 4% soiling loss (monthly cleaning assumed)

shading\_factor = 0.98 # 2% shading loss (well-designed rooftop)

reflection\_factor = 0.97 # 3% reflection loss (unchanged)

mismatch\_factor = 0.98 # 2% mismatch loss (modern panels)

dc\_wiring\_factor = 0.99 # 1% DC wiring loss (optimized design)

ac\_wiring\_factor = 0.995 # 0.5% AC wiring loss (short runs)

pre\_temp\_efficiency = (soiling\_factor \* shading\_factor \* reflection\_factor \*

mismatch\_factor \* dc\_wiring\_factor \* ac\_wiring\_factor)

logging.info(f"REALISTIC Loss Model for Athens:")

logging.info(f" Soiling losses: {(1-soiling\_factor)\*100:.1f}% (dust, pollution)")

logging.info(f" Shading losses: {(1-shading\_factor)\*100:.1f}% (rooftop installation)")

logging.info(f" Reflection losses: {(1-reflection\_factor)\*100:.1f}%")

logging.info(f" Mismatch losses: {(1-mismatch\_factor)\*100:.1f}%")

logging.info(f" DC wiring losses: {(1-dc\_wiring\_factor)\*100:.1f}%")

logging.info(f" AC wiring losses: {(1-ac\_wiring\_factor)\*100:.1f}%")

logging.info(f" Combined pre-temp efficiency: {pre\_temp\_efficiency:.3f} ({pre\_temp\_efficiency\*100:.1f}%)")

# Step 1: Calculate incident energy

df['incident\_irradiance'] = df['total\_irradiance'] # W/m²

df['incident\_energy'] = df['incident\_irradiance'] \* total\_panel\_area \* TIME\_INTERVAL\_HOURS # Wh

df['E\_incident'] = df['incident\_energy']

# Step 2: Cell temperature calculation (more accurate for Athens climate)

df['cell\_temperature'] = df['Air Temp'] + (NOCT - 20) \* df['incident\_irradiance'] / 800

# Step 3: Temperature factor

df['temperature\_factor'] = 1 + TEMP\_COEFF\_PMAX \* (df['cell\_temperature'] - 25)

df['temperature\_factor'] = df['temperature\_factor'].clip(lower=0)

# Step 4: DC power calculations

df['dc\_power\_stc'] = df['incident\_irradiance'] \* total\_panel\_area \* panel\_efficiency\_stc

df['dc\_power\_with\_losses'] = df['dc\_power\_stc'] \* pre\_temp\_efficiency

df['dc\_power\_actual'] = df['dc\_power\_with\_losses'] \* df['temperature\_factor']

df['dc\_power\_actual'] = df['dc\_power\_actual'].clip(upper=total\_system\_power\_stc)

# Step 5: AC power output (realistic inverter efficiency)

inverter\_efficiency = min(inverter\_params['eta\_inv\_nom'] / 100, 0.965) # Cap at 96%

df['ac\_power\_output'] = df['dc\_power\_actual'] \* inverter\_efficiency

inverter\_max\_ac = total\_system\_power\_stc \* inverter\_efficiency

df['ac\_power\_output'] = df['ac\_power\_output'].clip(upper=inverter\_max\_ac)

# Step 6: Energy calculations

df['E\_dc\_ideal'] = df['dc\_power\_stc'] \* TIME\_INTERVAL\_HOURS

df['E\_dc\_actual'] = df['dc\_power\_actual'] \* TIME\_INTERVAL\_HOURS

df['E\_ac'] = df['ac\_power\_output'] \* TIME\_INTERVAL\_HOURS

# Create ALL loss columns for compatibility

df['E\_loss\_pre\_temperature'] = df['incident\_energy'] \* panel\_efficiency\_stc - df['E\_dc\_ideal']

df['E\_loss\_temperature'] = df['E\_dc\_ideal'] - df['E\_dc\_actual']

df['E\_loss\_inverter'] = df['E\_dc\_actual'] - df['E\_ac']

df['E\_loss\_total'] = (df['incident\_energy'] \* panel\_efficiency\_stc) - df['E\_ac']

# Step 7: Performance Ratio

system\_power\_stc\_kw = total\_system\_power\_stc / 1000

df['reference\_yield'] = df['incident\_irradiance'] \* TIME\_INTERVAL\_HOURS / 1000

df['array\_yield'] = (df['E\_ac'] / 1000) / system\_power\_stc\_kw

df['PR'] = np.where(

df['reference\_yield'] > 0.01,

df['array\_yield'] / df['reference\_yield'],

0

)

df['PR'] = df['PR'].clip(0, 1.2)

# Additional columns for compatibility

df['dc\_power\_output\_per\_panel'] = df['dc\_power\_actual'] / number\_of\_panels

df['dc\_power\_output'] = df['dc\_power\_actual']

# IMPROVED: Validation with realistic bounds for Athens

valid\_pr = df[df['reference\_yield'] > 0.1]['PR']

if len(valid\_pr) > 0:

avg\_pr = valid\_pr.mean()

annual\_production = df['E\_ac'].sum() / 1000

specific\_yield = annual\_production / system\_power\_stc\_kw

logging.info(f"IMPROVED Energy Production Results:")

logging.info(f" Average PR: {avg\_pr:.3f} ({avg\_pr\*100:.1f}%)")

logging.info(f" Annual Production: {annual\_production:,.0f} kWh")

logging.info(f" Specific Yield: {specific\_yield:.0f} kWh/kWp")

logging.info(f" Pre-temp Efficiency: {pre\_temp\_efficiency\*100:.1f}%")

logging.info(f" System Size: {system\_power\_stc\_kw:.1f} kWp")

# Validate against realistic ranges for Athens

athens\_pr\_range = (0.76, 0.82) # Realistic PR range for Athens

athens\_yield\_range = (1350, 1550) # kWh/kWp/year for Athens

if athens\_pr\_range[0] <= avg\_pr <= athens\_pr\_range[1]:

logging.info(f"✓ PR validation OK: {avg\_pr:.3f} is within realistic range for Athens")

else:

logging.warning(f"⚠ PR {avg\_pr:.3f} is outside typical range {athens\_pr\_range} for Athens")

if athens\_yield\_range[0] <= specific\_yield <= athens\_yield\_range[1]:

logging.info(f"✓ Specific yield validation OK: {specific\_yield:.0f} kWh/kWp is realistic for Athens")

else:

logging.warning(f"⚠ Specific yield {specific\_yield:.0f} kWh/kWp is outside typical range {athens\_yield\_range} for Athens")

if specific\_yield > athens\_yield\_range[1]:

logging.warning(" Consider checking irradiance data or loss factors")

except Exception as e:

logging.error(f"Error calculating energy production: {e}", exc\_info=True)

raise

return df

## 4.5 Weighting and Consumption Functions

### 4.5.1 Weighting Factor Calculation

The calculate\_weighting\_factors function uses the solar zenith angle to determine a weighting factor for building usage during the day, starting by determining whether it is daylight, which is when the solar zenith angle is less than 90 degrees. To provide a weighting factor between 0 and 1, it normalizes the building's electricity use (load) during these daylight hours, by subtracting the minimum load value, and dividing the range of loads (the difference between the maximum and minimum values), for the daylight time, thus producing a scale in which 0 represents the lowest consumption and 1 represents the maximum consumption.

The weighting factor for each corresponding hour of the day is then determined by the normalized consumption, allocating a weighting factor of 1 for all daylight hours if the building's consumption is constant (that is, all values are the same), and during night, when solar energy production is not feasible, the weighting factor is set to zero. Because nocturnal hours do not contribute to the generation of solar energy, they are not included in the weighting process, guaranteeing that higher consumption during the day is given more weight, reflecting a greater need for energy, thus aiding in giving priority to energy storage and use throughout the day, when the building uses more energy.

1. Normalized Load (for adaptive\_improved strategy)

This scales the load values during daylight hours to a 0–1 range.

2. Time Factor (Gaussian curve centered at 13:00)

A bell-shaped curve peaking at 1 PM to emphasize midday hours.

3. Normalized Time Factor (0–1 range)

​

4. Solar Irradiance Normalization

If irradiance is present, it’s scaled from 0–1. Else defaults to 0.5.

5. Combined Weight (Weighted sum of components)

6. Final Weighting Factor (Scaled to 0.1–1.0)

7. Pure Load Matching (Optional strategy)

8. Peak Focused Weighting

9. Weighted Energy Production

def calculate\_weighting\_factors(df, strategy='adaptive\_improved'):

"""

Calculate weighting factors - ENHANCED with better validation.

"""

for col in ['Load (kW)', 'zenith']:

if col not in df.columns:

logging.error(f"'{col}' column is missing in the DataFrame.")

raise ValueError(f"'{col}' column is missing in the DataFrame.")

daylight\_mask = df['zenith'] < 90

df['weighting\_factor'] = 0.0

if not daylight\_mask.any():

logging.warning("No daylight hours detected; weighting factors remain zero.")

return df['weighting\_factor']

daylight\_load = df.loc[daylight\_mask, 'Load (kW)']

hours = df.loc[daylight\_mask].index.hour

# ENHANCED: Validate load data

if len(daylight\_load) == 0:

logging.warning("No load data available during daylight hours")

return df['weighting\_factor']

# Check for reasonable load values

if daylight\_load.min() < 0:

logging.warning(f"Negative load values detected: min = {daylight\_load.min():.1f} kW")

if daylight\_load.max() > 1000: # Adjust threshold as needed

logging.warning(f"Very high load values detected: max = {daylight\_load.max():.1f} kW")

if strategy == 'adaptive\_improved':

if len(daylight\_load) > 1:

min\_load = daylight\_load.min()

max\_load = daylight\_load.max()

if max\_load != min\_load:

normalized\_load = (daylight\_load - min\_load) / (max\_load - min\_load)

else:

normalized\_load = pd.Series(0.5, index=daylight\_load.index)

logging.warning("Load values are constant - using uniform weighting")

# FIXED: Time factor calculation with better centering

time\_factor = np.exp(-0.5 \* ((hours - 13) / 4) \*\* 2) # Peak at 13h (1 PM)

time\_factor = pd.Series(time\_factor, index=daylight\_load.index)

if time\_factor.max() != time\_factor.min():

time\_factor = (time\_factor - time\_factor.min()) / (time\_factor.max() - time\_factor.min())

else:

time\_factor = pd.Series(0.5, index=daylight\_load.index)

solar\_irradiance = df.loc[daylight\_mask, 'SolRad\_Hor']

if solar\_irradiance.max() > 0:

solar\_factor = solar\_irradiance / solar\_irradiance.max()

else:

solar\_factor = pd.Series(0.5, index=daylight\_load.index)

combined\_weight = (0.6 \* normalized\_load +

0.2 \* time\_factor +

0.2 \* solar\_factor)

if combined\_weight.max() != combined\_weight.min():

combined\_weight = (combined\_weight - combined\_weight.min()) / (combined\_weight.max() - combined\_weight.min())

final\_weights = 0.1 + 0.9 \* combined\_weight

df.loc[daylight\_mask, 'weighting\_factor'] = final\_weights

elif strategy == 'pure\_load\_matching':

if daylight\_load.max() != daylight\_load.min():

weights = (daylight\_load - daylight\_load.min()) / (daylight\_load.max() - daylight\_load.min())

weights = 0.2 + 0.8 \* weights

else:

weights = pd.Series(0.5, index=daylight\_load.index)

logging.warning("Constant load - using uniform weights for pure\_load\_matching")

df.loc[daylight\_mask, 'weighting\_factor'] = weights

elif strategy == 'peak\_focused':

base\_weight = 0.2

if daylight\_load.max() != daylight\_load.min():

load\_norm = (daylight\_load - daylight\_load.min()) / (daylight\_load.max() - daylight\_load.min())

else:

load\_norm = pd.Series(0.5, index=daylight\_load.index)

peak\_threshold = daylight\_load.quantile(0.8)

peak\_multiplier = pd.Series(1.0, index=daylight\_load.index)

peak\_multiplier[daylight\_load >= peak\_threshold] = 2.0

weights = base\_weight + 0.8 \* load\_norm \* peak\_multiplier

df.loc[daylight\_mask, 'weighting\_factor'] = weights

# ENHANCED: Validation of weighting factors

avg\_weight = df.loc[daylight\_mask, 'weighting\_factor'].mean()

min\_weight = df.loc[daylight\_mask, 'weighting\_factor'].min()

max\_weight = df.loc[daylight\_mask, 'weighting\_factor'].max()

logging.info(f"Weighting strategy '{strategy}':")

logging.info(f" Average weight: {avg\_weight:.3f}")

logging.info(f" Weight range: {min\_weight:.3f} to {max\_weight:.3f}")

logging.info(f" Daylight hours with weights: {daylight\_mask.sum()}")

return df['weighting\_factor']

def calculate\_weighted\_energy(df):

"""Calculate weighted energy production."""

for col in ['E\_ac', 'weighting\_factor']:

if col not in df.columns:

logging.error(f"'{col}' column is missing in the DataFrame.")

raise ValueError(f"'{col}' column is missing in the DataFrame.")

weighted\_energy = (df['E\_ac'] \* df['weighting\_factor']).sum() / 1000 # Convert Wh to kWh

return weighted\_energy

## 4.6 Multi-Objective Optimization Implementation

### 4.6.1 Objective Function

In multi-objective optimization, the objective\_function\_multi is a two-objective function that is used to maximize total energy output and reduce total weighted energy mismatch, accepting two input angles: tilt and azimuth along with a data subset (df\_subset), the extraterrestrial DNI (dni\_extra), the number of panels, and inverter settings.

To ensure that the optimization algorithm avoids invalid solutions, the function first determines whether the tilt angle (0° to 90°) and azimuth angle (90° to 270°) are within the acceptable range. If the angles are out of range, it returns large penalty values (infinity for mismatch and negative infinity for energy production), with the function then using the calculate\_total\_irradiance and calculate\_energy\_production functions to determine the energy production after calculating the total irradiance on the tilted surface. Using the calculate\_weighting\_factors function, the discrepancy between the energy generated (AC power output) and the building's load is computed, weighted by the consumption during the day, and then multiplying the weighting factor by the absolute magnitude of the mismatch and then use it to get the total weighted mismatch. In addition, kWh is calculated by adding up all of the energy production, and then returned by the function: maximizing overall energy output and reducing the total weighted mismatch, with the NSGA-II optimization algorithm producing a Pareto front of optimal trade-offs between two conflicting objectives thanks to this dual-objective structure, which aids in determining the ideal tilt and azimuth angles that strike a balance between energy supply and consumption requirements.

def objective\_function\_multi(angles, df\_subset, dni\_extra, number\_of\_panels, inverter\_params,

weighting\_strategy='adaptive\_improved'):

"""

IMPROVED Multi-objective function using realistic energy production calculation.

Uses penalty approach instead of infinite returns for bounds violations.

Parameters:

- angles: [tilt\_angle, azimuth\_angle]

- df\_subset: DataFrame subset for calculations

- dni\_extra: Extra-terrestrial DNI values

- number\_of\_panels: Number of panels in system

- inverter\_params: Inverter parameters dictionary

- weighting\_strategy: Strategy for calculating load weighting factors

"""

try:

tilt\_angle, azimuth\_angle = angles

# Use penalty approach instead of returning infinite values

penalty = 0.0

# Apply bounds penalties but don't return inf

if tilt\_angle < 0:

penalty += abs(tilt\_angle) \* 1000

elif tilt\_angle > 90:

penalty += (tilt\_angle - 90) \* 1000

# Azimuth bounds (0-360°) - ALLOW SOUTH (180°)

if azimuth\_angle < 0:

penalty += abs(azimuth\_angle) \* 100

elif azimuth\_angle > 360:

penalty += (azimuth\_angle - 360) \* 100

# Discourage North-facing orientations (270-90°) for Athens

if 270 <= azimuth\_angle <= 360 or 0 <= azimuth\_angle <= 90:

north\_penalty = min(90 - abs(azimuth\_angle - 360 if azimuth\_angle > 180 else azimuth\_angle), 90)

penalty += north\_penalty \* 10 # Mild penalty for north-facing

# Calculate performance with corrected angles - USING IMPROVED FUNCTION

df\_temp = df\_subset.copy()

df\_temp = calculate\_total\_irradiance(df\_temp, tilt\_angle, azimuth\_angle, dni\_extra)

df\_temp = calculate\_energy\_production(df\_temp, number\_of\_panels, inverter\_params) # FIXED

total\_energy\_production = df\_temp['E\_ac'].sum() / 1000 # kWh

# Use better weighting strategy

df\_temp['weighting\_factor'] = calculate\_weighting\_factors(df\_temp, strategy=weighting\_strategy)

df\_temp['load\_wh'] = df\_temp['Load (kW)'] \* 1000 \* TIME\_INTERVAL\_HOURS

df\_temp['hourly\_mismatch'] = df\_temp['E\_ac'] - df\_temp['load\_wh']

df\_temp['weighted\_mismatch'] = df\_temp['weighting\_factor'] \* np.abs(df\_temp['hourly\_mismatch'] / 1000)

total\_weighted\_mismatch = df\_temp['weighted\_mismatch'].sum()

# Add boundary violation penalties to mismatch

adjusted\_mismatch = total\_weighted\_mismatch + penalty

if not np.isfinite(adjusted\_mismatch):

adjusted\_mismatch = 1e6

if not np.isfinite(total\_energy\_production):

total\_energy\_production = 0.0

# Scale objectives to similar ranges for better NSGA-II performance

normalized\_mismatch = adjusted\_mismatch / 1000 # Scale to ~0.01-1.0 range

normalized\_production = total\_energy\_production / 1000 # Scale to ~2-4 range

return (normalized\_mismatch, normalized\_production)

except Exception as e:

logging.error(f"Error in objective\_function\_multi: {e}", exc\_info=True)

return (1e6, 0.0)

### 4.6.2 Optimization Context & Worker Management

The optimization process is made more efficient by the OptimizationContext class and the worker management system that goes along with it, especially in environments that use parallel processing. Important information utilized during the optimization process, including the data subset (df\_subset), extraterrestrial DNI (dni\_extra), number of panels, and inverter settings, is stored in the OptimizationContext class, facilitating effective access to common data and lowers computational overhead by guaranteeing that the same data is not transferred or computed repeatedly.

Each worker process in a parallel environment is initialized by the init\_worker function, which also sets the global optimization\_context to the worker's scope, reducing redundant data transfers across several processes, with the evaluation function for the DEAP optimization framework being evaluate\_individual, assessing each potential solution (a collection of tilt and azimuth angles), invoking the objective\_function\_multi after retrieving the relevant information from the common context, thus greatly expediting the optimization process and increasing efficiency by enabling parallel processing, which speeds up the evaluation of fitness values for a population of candidate solutions.

class OptimizationContext:

def \_\_init\_\_(self, df\_subset, dni\_extra, number\_of\_panels, inverter\_params):

self.df\_subset = df\_subset

self.dni\_extra = dni\_extra

self.number\_of\_panels = number\_of\_panels

self.inverter\_params = inverter\_params

# Global variable for multiprocessing

optimization\_context = None

def init\_worker(context):

"""Initialize worker process with shared context."""

global optimization\_context

optimization\_context = context

def evaluate\_individual(individual):

"""Top-level evaluation function for DEAP."""

global optimization\_context

return objective\_function\_multi(

individual,

optimization\_context.df\_subset,

optimization\_context.dni\_extra,

optimization\_context.number\_of\_panels,

optimization\_context.inverter\_params

)

### 4.6.3 DEAP Optimization Setup

The NSGA-II algorithm from the DEAP framework is used to set up and execute a multi-objective optimization using the run\_deap\_multi\_objective\_optimization function, aiming to find the ideal balance between maximizing energy production and decreasing the overall weighted energy mismatch. In order to exchange important information throughout the optimization process, such as the data subset, DNI extra, panel count, and inverter parameters, it first dynamically sets default values for population size and generations and then generates an OptimizationContext.

The code first creates a custom FitnessMulti class and an Individual class to build up DEAP's multi-objective optimization setup, serving to specify the two goals, which are to maximize energy output and minimize the weighted energy mismatch. To create new individuals, genetic operators such as blend crossover (cxBlend) and Gaussian mutation (mutGaussian) are registered. Initializing a multiprocessing pool allows for parallelization, which significantly boosts performance by evaluating people simultaneously, with a custom NSGA-II loop (custom\_nsga2\_evolution) managing the evolution process, and important metrics like average, standard deviation, and minimum/maximum fitness values are monitored over the course of the generations. Following the completion of the evolutionary process, the best-balanced solution is chosen after the results are filtered to find options that satisfy a specific energy production threshold, and finally making a thorough examination of the best trade-offs between the two goals by saving the Pareto front, filtering results, and choosing a solution.

def run\_deap\_multi\_objective\_optimization(

df\_subset,

dni\_extra,

number\_of\_panels,

inverter\_params,

output\_dir,

latitude,

longitude,

pop\_size=None,

max\_gen=None,

weighting\_strategy='adaptive\_improved'

):

"""

IMPROVED multi-objective optimization with realistic energy calculations.

"""

# Dynamic defaults for pop\_size / max\_gen

if pop\_size is None:

pop\_size = 50

if max\_gen is None:

max\_gen = 30

logging.info("=== IMPROVED Multi-objective optimization with realistic energy model ===")

# Create an optimization context object

context = OptimizationContext(

df\_subset=df\_subset,

dni\_extra=dni\_extra,

number\_of\_panels=number\_of\_panels,

inverter\_params=inverter\_params,

latitude=latitude,

longitude=longitude,

weighting\_strategy=weighting\_strategy

)

# Set up DEAP 'creator' for 2-objective optimization

if not hasattr(creator, "FitnessMulti"):

creator.create("FitnessMulti", base.Fitness, weights=(-1.0, 1.0))

if not hasattr(creator, "Individual"):

creator.create("Individual", list, fitness=creator.FitnessMulti)

# Toolbox with bounded operators

toolbox = base.Toolbox()

# Bounded individual creation

def create\_bounded\_individual():

"""Create individual within valid bounds"""

tilt = random.uniform(0, 90)

azimuth = random.uniform(0, 360)

return creator.Individual([tilt, azimuth])

# Register functions with bounds - USING IMPROVED EVALUATION

toolbox.register("individual", create\_bounded\_individual)

toolbox.register("population", tools.initRepeat, list, toolbox.individual)

toolbox.register("evaluate", evaluate\_individual\_IMPROVED) # FIXED

# Use bounded operators

toolbox.register("mate", tools.cxSimulatedBinaryBounded,

eta=20.0, low=[0, 0], up=[90, 360])

toolbox.register("mutate", tools.mutPolynomialBounded,

eta=20.0, low=[0, 0], up=[90, 360], indpb=0.1)

toolbox.register("select", tools.selNSGA2)

# Add bounds checking decorator as safety net

def checkBounds(min\_vals, max\_vals):

def decorator(func):

def wrapper(\*args, \*\*kwargs):

offspring = func(\*args, \*\*kwargs)

for child in offspring:

for i in range(len(child)):

child[i] = np.clip(child[i], min\_vals[i], max\_vals[i])

return offspring

return wrapper

return decorator

toolbox.decorate("mate", checkBounds([0, 0], [90, 360]))

toolbox.decorate("mutate", checkBounds([0, 0], [90, 360]))

# Optional: Parallelize

pool = multiprocessing.Pool(initializer=init\_worker, initargs=(context,))

toolbox.register("map", pool.map)

# Initialize population

population = toolbox.population(n=pop\_size)

# Prepare statistics and HallOfFame

stats = tools.Statistics(lambda ind: ind.fitness.values)

stats.register("avg", np.mean, axis=0)

stats.register("std", np.std, axis=0)

stats.register("min", np.min, axis=0)

stats.register("max", np.max, axis=0)

hof = tools.ParetoFront()

# Run the global custom NSGA-II evolution loop

final\_pop, logbook = custom\_nsga2\_evolution(

population,

toolbox,

cxpb=0.7,

mutpb\_start=0.2,

mutpb\_end=0.05,

ngen=max\_gen,

stats=stats,

halloffame=hof

)

# Close and join the multiprocessing pool

pool.close()

pool.join()

# Extract final Pareto front and perform post-processing

pareto\_front = list(hof)

# Filter solutions based on a production threshold (ADJUSTED for realistic values)

system\_capacity\_kw = (number\_of\_panels \* 240) / 1000

# More realistic threshold based on improved energy calculations

production\_threshold = system\_capacity\_kw \* 1400 / 1000 # 1400 kWh/kWp \* capacity in MW

logging.info(f"Production threshold set to {production\_threshold:.0f} MWh (realistic for Athens)")

filtered\_front = [ind for ind in pareto\_front if ind.fitness.values[1] >= production\_threshold]

logging.info(f"Filtered front: {len(filtered\_front)} of {len(pareto\_front)} pass production >= {production\_threshold:.0f}")

# Select the "most balanced" solution

if filtered\_front:

mismatch\_vals = np.array([ind.fitness.values[0] for ind in filtered\_front])

production\_vals = np.array([ind.fitness.values[1] for ind in filtered\_front])

# Normalize both objectives

mismatch\_norm = (mismatch\_vals - mismatch\_vals.min()) / (np.ptp(mismatch\_vals) + 1e-9)

prod\_norm = (production\_vals.max() - production\_vals) / (np.ptp(production\_vals) + 1e-9)

diff = np.abs(mismatch\_norm - prod\_norm)

best\_idx = np.argmin(diff)

best\_balanced = filtered\_front[best\_idx]

else:

best\_balanced = None

# Save the Pareto front to CSV

rows = []

for ind in pareto\_front:

mismatch, production = ind.fitness.values

rows.append({

'tilt\_angle': ind[0],

'azimuth\_angle': ind[1],

'weighted\_mismatch\_kWh': mismatch \* 1000, # Convert back from normalized

'total\_energy\_production\_kWh': production \* 1000, # Convert back from normalized

'azimuth\_deviation\_from\_south': abs(ind[1] - 180)

})

pareto\_df = pd.DataFrame(rows)

outfile = os.path.join(output\_dir, 'pareto\_front\_results.csv')

pareto\_df.to\_csv(outfile, index=False)

logging.info(f"Pareto front results saved to {outfile}")

return pareto\_front, filtered\_front, best\_balanced, logbook

### 4.6.4 Custom NSGA-II Evolution

By adjusting the mutation rate over generations, the custom\_nsga2\_evolution function modifies the NSGA-II evolutionary algorithm, and as the population converges, it progressively lowers this number to encourage refinement, starting with a larger mutation probability for wider exploration, with adaptive mutation, crossover, and tournament selection being used to create offspring, guaranteeing that novel solutions advance toward improved performance. The population is subjected to Pareto-based environmental selection, keeping the best individuals based on the dual goals of maximizing energy production and minimizing energy mismatch, with the Pareto front being preserved throughout the evolution in a hall-of-fame archive, and comprehensive statistics are documented to track convergence and advancement effectively optimizing the problem space while guaranteeing a balance between exploration and exploitation.

def custom\_nsga2\_evolution(pop, toolbox, cxpb, mutpb\_start, mutpb\_end, ngen, stats, halloffame):

"""

Custom NSGA-II evolution loop with adaptive mutation probability.

"""

logbook = tools.Logbook()

logbook.header = ['gen', 'nevals'] + (stats.fields if stats else [])

# Evaluate the initial population.

invalid\_ind = [ind for ind in pop if not ind.fitness.valid]

fitnesses = toolbox.map(toolbox.evaluate, invalid\_ind)

for ind, fit in zip(invalid\_ind, fitnesses):

ind.fitness.values = fit

pop = toolbox.select(pop, len(pop))

if halloffame is not None:

halloffame.update(pop)

record = stats.compile(pop) if stats else {}

logbook.record(gen=0, nevals=len(invalid\_ind), \*\*record)

# Log meaningful values

best\_ind = min(pop, key=lambda x: x.fitness.values[0])

logging.info(f"Generation 0 - best mismatch={best\_ind.fitness.values[0]:.2f}, best production={best\_ind.fitness.values[1]:.2f}")

# Evolution loop.

for gen in range(1, ngen + 1):

fraction = gen / float(ngen)

current\_mutpb = mutpb\_start + fraction \* (mutpb\_end - mutpb\_start)

offspring = varOr(pop, toolbox, lambda\_=len(pop), cxpb=cxpb, mutpb=current\_mutpb)

invalid\_ind = [ind for ind in offspring if not ind.fitness.valid]

fitnesses = toolbox.map(toolbox.evaluate, invalid\_ind)

for ind, fit in zip(invalid\_ind, fitnesses):

ind.fitness.values = fit

pop = toolbox.select(pop + offspring, k=len(pop))

if halloffame is not None:

halloffame.update(pop)

record = stats.compile(pop) if stats else {}

logbook.record(gen=gen, nevals=len(invalid\_ind), \*\*record)

best\_ind = min(pop, key=lambda x: x.fitness.values[0])

mismatch, production = best\_ind.fitness.values

logging.info(f"Gen {gen} - best mismatch={mismatch:.2f}, best prod={production:.2f}, mutpb={current\_mutpb:.3f}")

return pop, logbook

## 4.7 Seasonal Analysis Implementation

For a given dataset, the analyze\_seasonal\_performance function examines the seasonal changes in energy output, consumption, and matching, initially dividing the data into four seasons: winter, spring, summer, and fall based on the month of the year. It then computes important energy metrics for every season, including total energy produced (E\_ac), total load (load\_wh), surplus energy, energy deficit, and solar energy spent. Important ratios are also calculated by it, such as the self-sufficiency ratio (the percentage of solar energy consumed compared to total load) and the self-consumption ratio (the percentage of solar energy utilized compared to total production).

For ease of reading, the function transforms the findings into kWh and aggregates these measures by season, also determining daily averages per season, aiding in highlighting patterns over time. For clarity, the results are shown chronologically, with daily averages and seasonal data for every season, offering a thorough grasp of energy performance, while also identifiying possible areas for enhancement in solar energy usage and energy management throughout the year.

def analyze\_seasonal\_performance(df):

"""

Analyze seasonal variations in production, consumption, and matching.

"""

# Make a copy to avoid modifying the original dataframe

df\_season = df.copy()

# Add time-based columns

df\_season['month'] = df\_season.index.month

df\_season['day\_of\_year'] = df\_season.index.dayofyear

# Define seasons (Northern Hemisphere)

season\_mapping = {

1: 'Winter', 2: 'Winter', 3: 'Spring', 4: 'Spring', 5: 'Spring',

6: 'Summer', 7: 'Summer', 8: 'Summer', 9: 'Fall', 10: 'Fall',

11: 'Fall', 12: 'Winter'

}

df\_season['season'] = df\_season['month'].map(season\_mapping)

# Calculate mismatch between production and consumption

df\_season['load\_wh'] = df\_season['Load (kW)'] \* 1000 # Convert kW to W for comparison

df\_season['mismatch'] = df\_season['E\_ac'] - df\_season['load\_wh']

df\_season['surplus'] = df\_season['mismatch'].clip(lower=0)

df\_season['deficit'] = (-df\_season['mismatch']).clip(lower=0)

# Calculate self-consumption and self-sufficiency

df\_season['consumed\_solar'] = np.minimum(df\_season['E\_ac'], df\_season['load\_wh'])

# Aggregate metrics by season

seasonal\_stats = df\_season.groupby('season').agg({

'E\_ac': 'sum',

'load\_wh': 'sum',

'surplus': 'sum',

'deficit': 'sum',

'consumed\_solar': 'sum'

})

# Calculate additional metrics

seasonal\_stats['net\_balance\_wh'] = seasonal\_stats['E\_ac'] - seasonal\_stats['load\_wh']

seasonal\_stats['self\_consumption\_ratio'] = (seasonal\_stats['consumed\_solar'] / seasonal\_stats['E\_ac']) \* 100

seasonal\_stats['self\_sufficiency\_ratio'] = (seasonal\_stats['consumed\_solar'] / seasonal\_stats['load\_wh']) \* 100

# Convert to kWh for better readability

for col in ['E\_ac', 'load\_wh', 'surplus', 'deficit', 'consumed\_solar', 'net\_balance\_wh']:

seasonal\_stats[f'{col}\_kwh'] = seasonal\_stats[col] / 1000

# Calculate daily averages by season

df\_season['date\_str'] = df\_season.index.strftime('%Y-%m-%d')

# Group by season and date string, then calculate daily sums

daily\_sums = df\_season.groupby(['season', 'date\_str']).agg({

'E\_ac': 'sum',

'load\_wh': 'sum',

'consumed\_solar': 'sum'

})

# Then calculate the mean of these daily sums for each season

daily\_seasonal = daily\_sums.groupby('season').mean()

# Convert daily averages to kWh

for col in ['E\_ac', 'load\_wh', 'consumed\_solar']:

daily\_seasonal[f'{col}\_kwh'] = daily\_seasonal[col] / 1000

# Reorder seasons for chronological display

season\_order = ['Winter', 'Spring', 'Summer', 'Fall']

seasonal\_stats = seasonal\_stats.reindex(season\_order)

daily\_seasonal = daily\_seasonal.reindex(season\_order)

return seasonal\_stats, daily\_seasonal

## 4.8 Battery Sizing and Simulation

To find the most economical battery size for a certain energy system, the calculate\_optimal\_battery\_capacity function models battery performance across a range of capacities (from 1 to 100 kWh), assessing important parameters like self-consumption rate, self-sufficiency rate, battery charging and discharging behavior, and energy losses by simulating hourly energy flows, also taking into account practical battery attributes, such as depth of discharge (DoD) and efficiency, and evaluates both technical and financial performance using these metrics. In order to evaluate the profitability of the investment, the simulation also determines the battery's basic payback period.

The function determines the ideal battery capacity by taking into account both technical (like the self-sufficiency rate) and economic (such the payback period, annual savings from less grid imports, and revenue from grid exports) variables, with the function selecting the configuration with the lowest payback period if more than one satisfies the desired self-sufficiency criteria. Additionally, it offers comprehensive analysis and visualizations of the battery's performance, assisting users in selecting the ideal battery size for their requirements, integrating technical and economic viewpoints in order to maximize battery sizing for both financial viability and energy efficiency.

def calculate\_energy\_based\_battery\_capacity(df: pd.DataFrame,

output\_dir: str,

target\_self\_sufficiency: float = 80.0,

surplus\_capture\_target: float = 90.0,

battery\_round\_trip\_efficiency: float = 0.90,

depth\_of\_discharge: float = 0.80,

min\_capacity\_kwh: float = 2.5,

max\_capacity\_kwh: float = 1000.0) -> Tuple[float, Dict]: # Changed default to 1000

"""

Calculate optimal battery capacity based on energy patterns, surplus capture, and self-consumption.

Enhanced approach optimizes for:

1. High surplus capture (minimize wasted solar energy)

2. High self-consumption (minimize grid export during surplus periods)

3. Balanced performance between both metrics

4. Avoid oversized batteries with poor utilization

Parameters:

-----------

df : pd.DataFrame

DataFrame with 'E\_ac' (Wh) and 'Load (kW)' columns

output\_dir : str

Directory for output files

target\_self\_sufficiency : float

Target self-sufficiency percentage (default: 80%)

surplus\_capture\_target : float

Target percentage of surplus energy to capture (default: 90%)

battery\_round\_trip\_efficiency : float

Battery round-trip efficiency (default: 0.90)

depth\_of\_discharge : float

Battery depth of discharge (default: 0.80)

min\_capacity\_kwh : float

Minimum battery capacity to test (default: 2.5 kWh)

max\_capacity\_kwh : float

Maximum battery capacity to test (default: 1000 kWh)

Returns:

--------

Tuple[float, Dict]: (optimal\_capacity\_kwh, detailed\_results)

"""

logging.info("Starting energy-based battery optimization with surplus capture + self-consumption focus...")

# Step 1: Analyze energy patterns

energy\_analysis = analyze\_energy\_patterns(df)

# Step 2: Determine capacity range based on energy patterns - updated for higher max

if max\_capacity\_kwh is None:

# For high consumption scenarios, test much larger batteries

max\_capacity\_kwh = min(energy\_analysis['max\_daily\_surplus\_kwh'] \* 10, 1000)

max\_capacity\_kwh = max(max\_capacity\_kwh, 100) # Ensure reasonable minimum

logging.info(f"Testing battery capacities from {min\_capacity\_kwh} to {max\_capacity\_kwh:.1f} kWh")

logging.info(f"Optimization focus: Balanced surplus capture + self-consumption")

# Step 3: Test different battery capacities with finer granularity for large range

if max\_capacity\_kwh <= 100:

capacity\_step = 2.5 # Fine steps for small batteries

elif max\_capacity\_kwh <= 500:

capacity\_step = 5.0 # Medium steps

else:

capacity\_step = 10.0 # Coarser steps for very large batteries

capacities\_kwh = np.arange(min\_capacity\_kwh, max\_capacity\_kwh + capacity\_step, capacity\_step)

results = []

one\_way\_efficiency = np.sqrt(battery\_round\_trip\_efficiency)

logging.info(f"Testing {len(capacities\_kwh)} different battery capacities...")

for i, capacity\_kwh in enumerate(capacities\_kwh):

# Progress logging for large ranges

if i % 20 == 0:

logging.info(f" Testing capacity {i+1}/{len(capacities\_kwh)}: {capacity\_kwh:.1f} kWh")

# Simulate battery performance

battery\_metrics = simulate\_battery\_performance(

df, capacity\_kwh, one\_way\_efficiency, depth\_of\_discharge

)

# Calculate surplus capture rate

total\_surplus\_kwh = energy\_analysis['total\_surplus\_kwh']

surplus\_stored\_kwh = battery\_metrics['total\_charged\_kwh']

surplus\_capture\_rate = (surplus\_stored\_kwh / total\_surplus\_kwh \* 100) if total\_surplus\_kwh > 0 else 0

# Store results with enhanced metrics

result = {

'capacity\_kwh': capacity\_kwh,

'self\_sufficiency\_pct': battery\_metrics['self\_sufficiency\_rate'],

'self\_consumption\_pct': battery\_metrics['self\_consumption\_rate'],

'surplus\_capture\_pct': surplus\_capture\_rate,

'grid\_import\_kwh': battery\_metrics['grid\_import\_kwh'],

'grid\_export\_kwh': battery\_metrics['grid\_export\_kwh'],

'battery\_cycles\_per\_year': battery\_metrics['equivalent\_full\_cycles'],

'battery\_utilization\_pct': (battery\_metrics['equivalent\_full\_cycles'] \* 100) / 365,

'avg\_soc\_pct': battery\_metrics['avg\_soc\_percent'],

# New metrics for enhanced analysis

'total\_stored\_kwh': battery\_metrics['total\_charged\_kwh'],

'surplus\_wasted\_kwh': total\_surplus\_kwh - surplus\_stored\_kwh,

}

results.append(result)

# Step 4: Find optimal capacity using enhanced multi-criteria approach

results\_df = pd.DataFrame(results)

optimal\_capacity = find\_optimal\_battery\_size(

results\_df, target\_self\_sufficiency, surplus\_capture\_target, energy\_analysis

)

# Step 5: Generate detailed analysis for optimal capacity

optimal\_metrics = simulate\_battery\_performance(

df, optimal\_capacity, one\_way\_efficiency, depth\_of\_discharge, detailed=True

)

# Step 6: Create enhanced visualizations

create\_energy\_based\_battery\_plots(results\_df, optimal\_capacity, energy\_analysis, output\_dir)

# Step 7: Save results

results\_df.to\_csv(os.path.join(output\_dir, 'energy\_based\_battery\_analysis\_enhanced.csv'), index=False)

# Step 8: Compile final results with enhanced analysis

final\_results = {

'optimal\_capacity\_kwh': optimal\_capacity,

'energy\_analysis': energy\_analysis,

'capacity\_analysis': results\_df,

'optimal\_performance': optimal\_metrics,

'recommendations': generate\_battery\_recommendations(optimal\_capacity, optimal\_metrics, energy\_analysis),

'optimization\_method': 'balanced\_surplus\_capture\_and\_self\_consumption'

}

# Enhanced logging

optimal\_result = results\_df[results\_df['capacity\_kwh'] == optimal\_capacity].iloc[0]

logging.info(f"Enhanced energy-based optimization completed:")

logging.info(f" Optimal battery capacity: {optimal\_capacity:.1f} kWh")

logging.info(f" Expected self-sufficiency: {optimal\_metrics['self\_sufficiency\_rate']:.1f}%")

logging.info(f" Expected self-consumption: {optimal\_metrics['self\_consumption\_rate']:.1f}%")

logging.info(f" Expected surplus capture: {optimal\_result['surplus\_capture\_pct']:.1f}%")

logging.info(f" Annual surplus wasted: {optimal\_result['surplus\_wasted\_kwh']:,.0f} kWh")

logging.info(f" Battery utilization: {optimal\_result['battery\_utilization\_pct']:.1f}%")

return optimal\_capacity, final\_results

## 4.9 Economic Analysis Implementation

With a focus on solar panel and battery installations, the economic analysis functions offer a comprehensive framework for assessing the technical and financial viability of renewable energy systems, with the upfront expenses, such as those for solar panel purchases, installation, inverters, battery storage, and balance of system (BOS) components, being determined using the calculate\_initial\_investment function. It offers a clear breakdown of the capital expenditure needed for system setup by evaluating the expenses related to each system component, with the calculate\_annual\_cashflow function then simulating yearly cash flows during the anticipated 25-year lifespan of the system, accounting for feed-in tariffs, grid exports, inflation, maintenance expenses, and energy savings from direct consumption, providing a realistic estimate of system performance and financial returns by including feed-in tariffs and power price hikes.

After that, the financial metrics, including important indicators like Net Present Value (NPV), Internal Rate of Return (IRR), Return on Investment (ROI), Payback Period, and Levelized Cost of Energy (LCOE), are computed in calculate\_financial\_metrics, giving a thorough picture of the investment's profitability and reveal information about its potential for cash flow and long-term viability. The calculate\_efficiency\_metrics function also assesses the system's operational efficiency, including the self-sufficiency and self-consumption ratios, system yield, capacity factor, and performance ratio. By finding the most efficient configurations, helping users evaluate how well the system generates and uses energy, which supports the overall economic analysis.

def calculate\_initial\_investment(number\_of\_panels, battery\_capacity\_kwh=None, panel\_cost=250,

installation\_cost\_per\_panel=150, inverter\_cost\_per\_kw=120,

battery\_cost\_per\_kwh=500, bos\_cost\_per\_panel=50):

"""Calculate the total initial investment costs."""

panel\_cost\_total = number\_of\_panels \* panel\_cost

installation\_cost = number\_of\_panels \* installation\_cost\_per\_panel

inverter\_cost = (number\_of\_panels \* 0.24) \* inverter\_cost\_per\_kw

battery\_cost = battery\_capacity\_kwh \* battery\_cost\_per\_kwh if battery\_capacity\_kwh else 0

bos\_cost = number\_of\_panels \* bos\_cost\_per\_panel

total\_investment = panel\_cost\_total + installation\_cost + inverter\_cost + battery\_cost + bos\_cost

return {

'panel\_cost': panel\_cost\_total,

'installation\_cost': installation\_cost,

'inverter\_cost': inverter\_cost,

'battery\_cost': battery\_cost,

'bos\_cost': bos\_cost,

'total\_investment': total\_investment

}

def calculate\_annual\_cashflow(df, electricity\_price=0.20, feed\_in\_tariff=0.10,

annual\_maintenance\_percent=0.5, inflation\_rate=2.0,

electricity\_price\_increase=3.0, system\_lifetime=25,

initial\_investment=None):

"""Calculate annual cash flows over the system lifetime."""

*# Calculate energy flows*

df\_calc = df.copy()

df\_calc['load\_wh'] = df\_calc['Load (kW)'] \* 1000

direct\_consumption = df\_calc.apply(lambda x: min(x['E\_ac'], x['load\_wh']), axis=1).sum()

grid\_exports = df\_calc.apply(lambda x: max(0, x['E\_ac'] - x['load\_wh']), axis=1).sum()

grid\_imports = df\_calc.apply(lambda x: max(0, x['load\_wh'] - x['E\_ac']), axis=1).sum()

*# Convert to kWh and calculate first year values*

direct\_consumption\_kwh = direct\_consumption / 1000

grid\_exports\_kwh = grid\_exports / 1000

grid\_imports\_kwh = grid\_imports / 1000

savings\_from\_direct\_use = direct\_consumption\_kwh \* electricity\_price

income\_from\_exports = grid\_exports\_kwh \* feed\_in\_tariff

annual\_maintenance\_cost = initial\_investment['total\_investment'] \* (annual\_maintenance\_percent / 100)

*# Calculate annual cash flows with degradation, inflation, and price increases*

cashflows = []

for year in range(1, system\_lifetime + 1):

*# Apply annual adjustments*

current\_electricity\_price = electricity\_price \* ((1 + electricity\_price\_increase/100) \*\* (year-1))

current\_feed\_in\_tariff = feed\_in\_tariff \* ((1 + inflation\_rate/100) \*\* (year-1))

current\_maintenance = annual\_maintenance\_cost \* ((1 + inflation\_rate/100) \*\* (year-1))

degradation\_factor = (1 - 0.005) \*\* (year-1)

*# Calculate year's cash flow*

year\_savings = direct\_consumption\_kwh \* degradation\_factor \* current\_electricity\_price

year\_income = grid\_exports\_kwh \* degradation\_factor \* current\_feed\_in\_tariff

year\_cashflow = year\_savings + year\_income - current\_maintenance

cashflows.append({

'year': year,

'savings': year\_savings,

'income': year\_income,

'maintenance': current\_maintenance,

'net\_cashflow': year\_cashflow

})

return pd.DataFrame(cashflows)

def calculate\_financial\_metrics(initial\_investment, cashflows, discount\_rate=5.0,

electricity\_price=0.20, electricity\_price\_increase=3.0,

feed\_in\_tariff=0.10, inflation\_rate=2.0):

"""Calculate key financial metrics: NPV, IRR, ROI, and payback period."""

investment = initial\_investment['total\_investment']

annual\_cashflows = cashflows['net\_cashflow'].tolist()

*# Calculate NPV*

npv = -investment

for i, cf in enumerate(annual\_cashflows):

npv += cf / ((1 + discount\_rate/100) \*\* (i+1))

*# Calculate IRR*

try:

irr = npv\_to\_irr([-investment] + annual\_cashflows)

except:

irr = None

*# Calculate ROI*

total\_returns = sum(annual\_cashflows)

roi = (total\_returns - investment) / investment \* 100

*# Calculate payback period*

cumulative\_cashflow = -investment

payback\_period = None

for i, cf in enumerate(annual\_cashflows):

cumulative\_cashflow += cf

if cumulative\_cashflow >= 0 and payback\_period is None:

if i > 0:

prev\_cf = cumulative\_cashflow - cf

fraction = -prev\_cf / cf

payback\_period = i + fraction

else:

payback\_period = i + 1

*# Calculate LCOE*

total\_production\_kwh = sum([

cashflows.iloc[i]['savings'] / (electricity\_price \* ((1 + electricity\_price\_increase/100) \*\* i)) +

cashflows.iloc[i]['income'] / (feed\_in\_tariff \* ((1 + inflation\_rate/100) \*\* i))

for i in range(len(cashflows))

])

*# Apply degradation*

total\_production\_kwh = total\_production\_kwh \* sum([(1 - 0.005) \*\* i for i in range(len(cashflows))]) / len(cashflows)

lcoe = investment / total\_production\_kwh if total\_production\_kwh > 0 else float('inf')

return {

'NPV': npv,

'IRR': irr \* 100 if irr is not None else None,

'ROI': roi,

'Payback\_Period\_Years': payback\_period,

'LCOE': lcoe

}

def calculate\_efficiency\_metrics(df, number\_of\_panels, panel\_area, total\_panel\_area, panel\_nominal\_power):

"""Calculate comprehensive system efficiency metrics."""

df\_copy = df.copy()

df\_copy['load\_wh'] = df\_copy['Load (kW)'] \* 1000

*# Calculate energy totals*

total\_incident\_energy\_kwh = df\_copy['E\_incident'].sum() / 1000

total\_produced\_ac\_kwh = df\_copy['E\_ac'].sum() / 1000

total\_consumption\_kwh = df\_copy['load\_wh'].sum() / 1000

direct\_consumption\_kwh = df\_copy.apply(lambda x: min(x['E\_ac'], x['load\_wh']), axis=1).sum() / 1000

*# Calculate efficiency metrics*

panel\_efficiency = panel\_nominal\_power / (panel\_area \* 1000)

system\_efficiency = total\_produced\_ac\_kwh / total\_incident\_energy\_kwh

system\_yield = total\_produced\_ac\_kwh / (number\_of\_panels \* panel\_nominal\_power / 1000)

*# Performance ratio calculation*

total\_irradiance\_per\_sqm = total\_incident\_energy\_kwh / total\_panel\_area

theoretical\_max\_energy = total\_irradiance\_per\_sqm \* panel\_efficiency \* total\_panel\_area

performance\_ratio = total\_produced\_ac\_kwh / theoretical\_max\_energy

*# Consumption metrics*

self\_consumption\_ratio = direct\_consumption\_kwh / total\_produced\_ac\_kwh if total\_produced\_ac\_kwh > 0 else 0

self\_sufficiency\_ratio = direct\_consumption\_kwh / total\_consumption\_kwh if total\_consumption\_kwh > 0 else 0

*# Capacity factor calculation (8760 hours per year)*

capacity\_factor = total\_produced\_ac\_kwh / (number\_of\_panels \* panel\_nominal\_power / 1000 \* 8760)

return {

'panel\_efficiency': panel\_efficiency \* 100,

'system\_efficiency': system\_efficiency \* 100,

'performance\_ratio': performance\_ratio \* 100,

'system\_yield': system\_yield,

'capacity\_factor': capacity\_factor \* 100,

'self\_consumption\_ratio': self\_consumption\_ratio \* 100,

'self\_sufficiency\_ratio': self\_sufficiency\_ratio \* 100

}

## 4.10 Main Execution Workflow

Starting with reading command-line arguments and configuring the required configurations for the analysis, the main() method coordinates the solar energy analysis tool's whole workflow, beginning by taking in a number of parameters, such as the output directory, configuration options, and file paths for the input data. Along with other available options including representative date, maximum iterations, and population size, it also manages the location's geographic details (latitude and longitude), making sure the output directory is there and configures logging for the analysis process after the inputs have been processed. Following the loading of the setup and input data, solar parameters including the sun's position and direct normal irradiance (DNI) are calculated, supporting the system optimization processes and are crucial for comprehending the energy that the solar panels can provide.

The function then defines system characteristics, such as the quantity of panels to be utilized depending on available area, the efficiency of the panels, and the parameters of the inverter, after doing the initial solar calculations, with the optimal system configuration being then found through a multi-objective optimization approach. Then, in order to simulate energy generation, the ideal tilt and azimuth angles for the panels are determined, carrying additional analyses, encompassing battery sizing, seasonal performance, and economic assessments that consider variables like feed-in tariffs, maintenance expenses, and electricity prices. To evaluate the system's economic feasibility, financial measures such as Net Present Value (NPV), Internal Rate of Return (IRR), and payback period are calculated, with the last step creating a thorough description of the findings, including energy production and investment data, and storing them in a CSV file for later study.

def main():

"""Main execution function."""

try:

*# Parse command-line arguments*

parser = argparse.ArgumentParser(description='Solar Energy Analysis Tool')

parser.add\_argument('--data\_file', type=str, required=True, help='Path to the input CSV data file')

parser.add\_argument('--output\_dir', type=str, required=True, help='Directory to save output')

parser.add\_argument('--config\_file', type=str, required=True, help='Path to YAML configuration')

parser.add\_argument('--latitude', type=float, default=37.98983, help='Latitude (default: Athens)')

parser.add\_argument('--longitude', type=float, default=23.74328, help='Longitude (default: Athens)')

parser.add\_argument('--representative\_date', type=str, default='2023-06-15',

help='Date for representative profiles')

parser.add\_argument('--maxiter', type=int, default=1000, help='Maximum iterations')

parser.add\_argument('--popsize', type=int, default=15, help='Population size')

args = parser.parse\_args()

*# Setup output directory and logging*

if not os.path.exists(args.output\_dir):

os.makedirs(args.output\_dir)

setup\_logging(args.output\_dir)

*# Load configuration and data*

config = load\_config(args.config\_file)

df\_original = load\_and\_preprocess\_data(args.data\_file)

*# Initial solar calculations*

df\_original = calculate\_solar\_position(df\_original, args.latitude, args.longitude)

df\_original = calculate\_dni(df\_original)

dni\_extra = pvlib.irradiance.get\_extra\_radiation(df\_original.index)

df\_subset = df\_original[['SolRad\_Hor', 'SolRad\_Dif', 'Air Temp', 'zenith', 'azimuth', 'DNI', 'Load (kW)']].copy()

*# Define system parameters*

number\_of\_panels = int(config.get('available\_area', 1500) //

((config['solar\_panel']['length'] + config['solar\_panel']['spacing\_length']) \*

(config['solar\_panel']['width'] + config['solar\_panel']['spacing\_width'])))

panel\_area = config['solar\_panel']['length'] \* config['solar\_panel']['width']

total\_panel\_area = panel\_area \* number\_of\_panels

panel\_efficiency = config['solar\_panel']['power\_rating'] / (panel\_area \* 1000)

inverter\_params = {'eta\_inv\_nom': config['inverter']['eta\_inv\_nom'],

'pdc0': config['solar\_panel']['pmp'] \* number\_of\_panels}

*# Run multi-objective optimization*

pareto\_front, filtered\_front, best\_balanced = run\_deap\_multi\_objective\_optimization(

df\_subset, dni\_extra, number\_of\_panels, inverter\_params, args.output\_dir,

pop\_size=args.popsize, max\_gen=args.maxiter

)

*# Extract optimal angles and simulate performance*

optimal\_tilt = best\_balanced[0]

optimal\_azimuth = best\_balanced[1]

balanced\_weighted\_mismatch, balanced\_production = best\_balanced.fitness.values

*# Calculate performance with optimal angles*

df = df\_subset.copy()

df = calculate\_total\_irradiance(df, optimal\_tilt, optimal\_azimuth, dni\_extra)

df = calculate\_energy\_production(df, number\_of\_panels, inverter\_params)

df['weighting\_factor'] = calculate\_weighting\_factors(df)

*# Seasonal analysis*

seasonal\_stats, daily\_seasonal = analyze\_seasonal\_performance(df)

*# Battery sizing analysis*

optimal\_capacity, battery\_results = calculate\_optimal\_battery\_capacity(

df, args.output\_dir, min\_capacity=1, max\_capacity=50, capacity\_step=2

)

*# Economic analysis*

initial\_investment = calculate\_initial\_investment(

number\_of\_panels, optimal\_capacity, panel\_cost=250, installation\_cost\_per\_panel=150,

inverter\_cost\_per\_kw=120, battery\_cost\_per\_kwh=500, bos\_cost\_per\_panel=50

)

cashflows = calculate\_annual\_cashflow(

df, electricity\_price=0.20, feed\_in\_tariff=0.10, annual\_maintenance\_percent=0.5,

inflation\_rate=2.0, electricity\_price\_increase=3.0, system\_lifetime=25,

initial\_investment=initial\_investment

)

financial\_metrics = calculate\_financial\_metrics(

initial\_investment, cashflows, discount\_rate=5.0, electricity\_price=0.20,

electricity\_price\_increase=3.0, feed\_in\_tariff=0.10, inflation\_rate=2.0

)

efficiency\_metrics = calculate\_efficiency\_metrics(

df, number\_of\_panels, panel\_area, total\_panel\_area,

config['solar\_panel']['power\_rating']

)

*# Create summary and save results*

summary = {

'Number of Panels Installed': number\_of\_panels,

'Total Panel Area (m²)': f"{total\_panel\_area:.2f}",

'Panel Efficiency (%)': f"{panel\_efficiency \* 100:.2f}",

'Optimal Tilt Angle (°)': f"{optimal\_tilt:.2f}",

'Optimal Azimuth Angle (°)': f"{optimal\_azimuth:.2f}",

'Balanced Weighted Energy Mismatch (kWh)': f"{balanced\_weighted\_mismatch:.2f}",

'Balanced Total Energy Produced (kWh)': f"{balanced\_production:.2f}",

*# ...additional metrics added here...*

'Total Investment ($)': f"{initial\_investment['total\_investment']:,.2f}",

'Net Present Value ($)': f"{financial\_metrics['NPV']:,.2f}",

'Internal Rate of Return (%)': f"{financial\_metrics['IRR']:.2f}",

'Payback Period (years)': f"{financial\_metrics['Payback\_Period\_Years']:.2f}",

'Levelized Cost of Electricity ($/kWh)': f"{financial\_metrics['LCOE']:.4f}",

'Performance Ratio (%)': f"{efficiency\_metrics['performance\_ratio']:.2f}",

'System Yield (kWh/kWp)': f"{efficiency\_metrics['system\_yield']:.2f}",

'Capacity Factor (%)': f"{efficiency\_metrics['capacity\_factor']:.2f}"

}

summary\_df = pd.DataFrame(list(summary.items()), columns=['Metric', 'Value'])

summary\_df.to\_csv(os.path.join(args.output\_dir, 'summary\_results.csv'), index=False, sep=';')

except Exception as e:

logging.error("An unexpected error occurred:", exc\_info=True)

sys.exit(1)

## 4.11 Forecasting Module Implementation

Using TensorFlow for deep learning and Pandas for data preparation, the forecasting module is constructed as a separate part of the system. In order to help operational decision-making and long-term infrastructure planning, its main function is to generate precise short and long-term energy consumption projections. Because of its capacity to capture long-range dependencies and temporal changes within hourly load profiles, the forecasting engine is based on an architecture known as Long Short-Term Memory (LSTM), where sequences of normalized consumption data are sent to the input layer, together with designed factors like the time of day, the day of the week, and the ambient temperature. To ensure strong generalization to unknown data, the network consists of LSTM layers followed by dense layers with dropout for regularization, with a single predicted load value each time step is provided by the final output layer.

Pandas is used to pre-process historical consumption data, including time-series resampling, outlier filtering, and dividing the data into training and validation sets using a rolling window technique. Early halting and learning rate decay are used to avoid overfitting, and the Adam optimization algorithm is used to train the model with Mean Squared Error (MSE) as the loss function, with a held-out portion of the dataset is being used to validate the model's performance, and measures like MAE, RMSE, and MAPE are used to measure accuracy.

The forecasting module uses standardized APIs to expose its prediction findings and functions as a service within the wider architecture. Its outputs are directly fed into the battery sizing module, which informs charge-discharge techniques, the economic analysis module, enabling precise cash flow estimates, and the multi-objective optimization engine, which informs scheduling constraints. The forecasting module serves as a vital link between data-driven prediction and practical operational planning by matching anticipated load profiles with generation and storage capacities.

A systematic, data-driven strategy is used to train the forecasting model, guaranteeing accuracy and generalization across the hourly load profiles, with the input dataset being composed of pre-processed hourly load data that has been enhanced with derived temporal and meteorological variables and arranged into fixed-length sequences In order to capture long-term temporal dependencies,. Prior to training, the dataset is split into three distinct sets:

* Training set (~70%), used for updating the network weights.
* Validation set (~15%), used for monitoring the model’s generalization ability and fine-tuning hyperparameters.
* Test set (~15%), held out for final performance evaluation.

An Early Stopping callback is used to enhance generalization and avoid overfitting, with the training procedure keeping track of the loss on the validation set, and it stops and saves the best-performing model after a predetermined number of epochs (patience) if no progress is seen. Additionally, as the validation loss reaches a plateau, a learning rate scheduler progressively lowers the learning rate, enabling the model to more accurately adjust its final weights. The trained model is rigorously assessed using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) after each input sequence has been normalized and modified to be compatible with the LSTM. This structured approach ensures that the forecasting model delivers both robust and accurate performance across varying load profiles and seasonal variations, making it an integral component of the overall PV optimization and forecasting workflow.

# 

# CHAPTER 5 Analysis

The main conclusions of the solar energy analysis carried out in the earlier phases are examined and interpreted in this chapter, with meaningful insights into the performance and viability of photovoltaic (PV) systems constructed using the technique given being extracted by critically analyzing the technical, financial, and operational outcomes, placing special focus on how the financial indicators, seasonal changes, energy production patterns, and optimization results match or deviate from the predicted theoretical behavior.

## 5.1 Visualization of Key Results

The results of the investigation have been interpreted with the use of a number of graphic outputs, consisting of financial estimates, battery sizing analyses, seasonal energy production profiles, and graphical depictions of optimization results, with the significant trends, system behaviors, and trade-offs found throughout the analysis visually summarized in the following graphs. Readers can better grasp the dynamic links between solar resource availability, system configuration options, energy performance, and economic viability by looking at the following charts and figures.

First we will examine the average energy values per hour in a day.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1: Weather, input and parameters data sample

Sample input data structure showing hourly load and weather parameters including electrical load (kW), air temperature (°C), solar irradiance components (W/m²), wind speed (m/s), and solar position angles (zenith, azimuth) used for system modeling.

#### Energy, Load & Irradiance Profiles

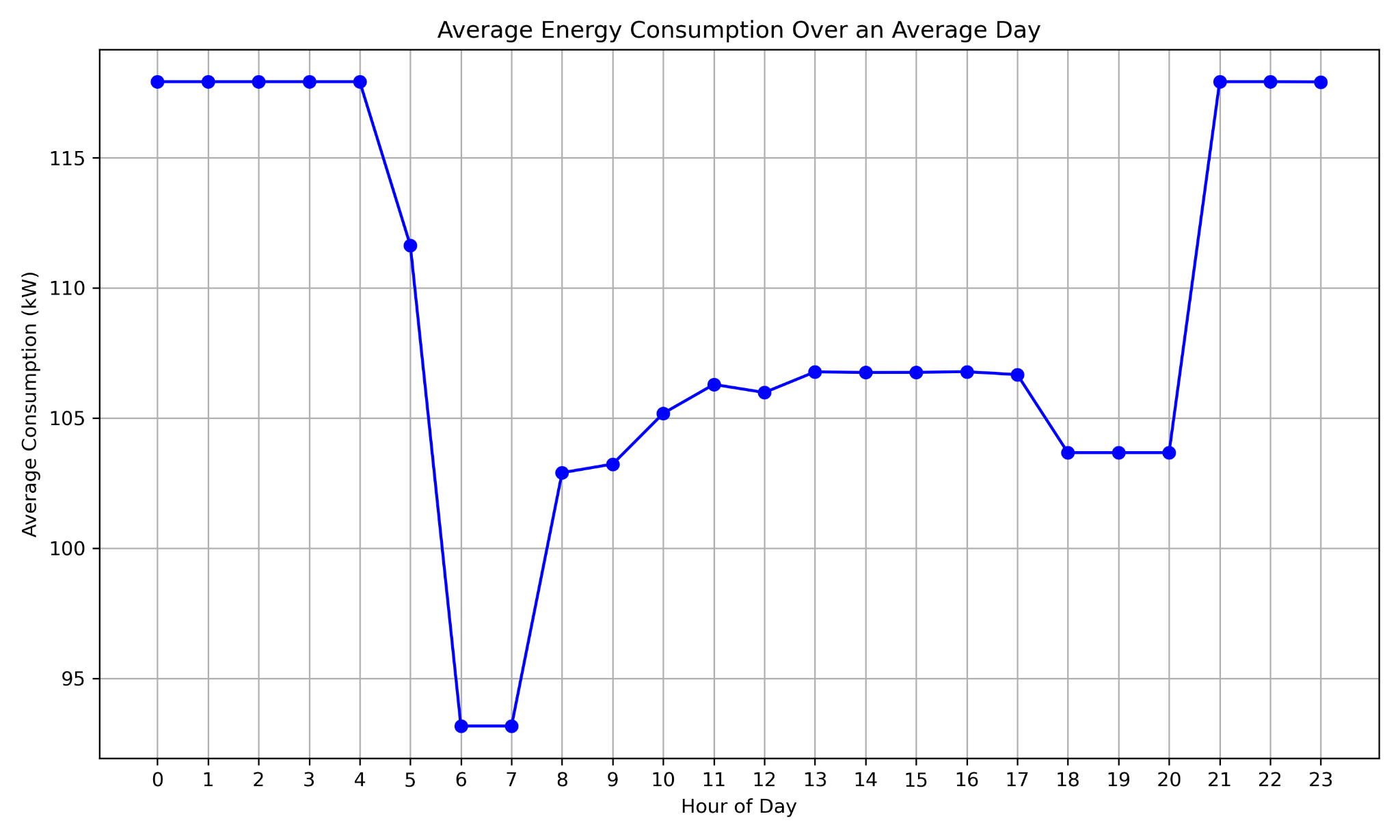


Figure 2: Average Energy Consumption over an average day

Daily average energy consumption profile showing building load patterns with peak consumption of 118 kW during nighttime hours and minimum demand of 93 kW during early morning (5-6 AM), illustrating typical residential/commercial energy usage patterns.

A group of colored bars

AI-generated content may be incorrect.

Figure 3: Top left: Monthly Average Solar Irradiance  
Top right: Monthly Total Solar Irradiation  
Bottom left: Monthly Average Building Load  
Bottom right: Monthly Average Temperature

Monthly summary charts presenting average solar irradiance, total solar irradiation, building load, and temperature patterns, highlighting the seasonal correlation between solar availability and cooling loads.

A graph showing a bar graph

AI-generated content may be incorrect.

Figure 4: Annual Building Profile - Athens

Annual building load profile for Athens showing electrical consumption patterns with peak demand of 126.8 kW, average load of 108.7 kW, and total annual consumption of 952,527 kWh, with higher summer loads due to cooling requirements.

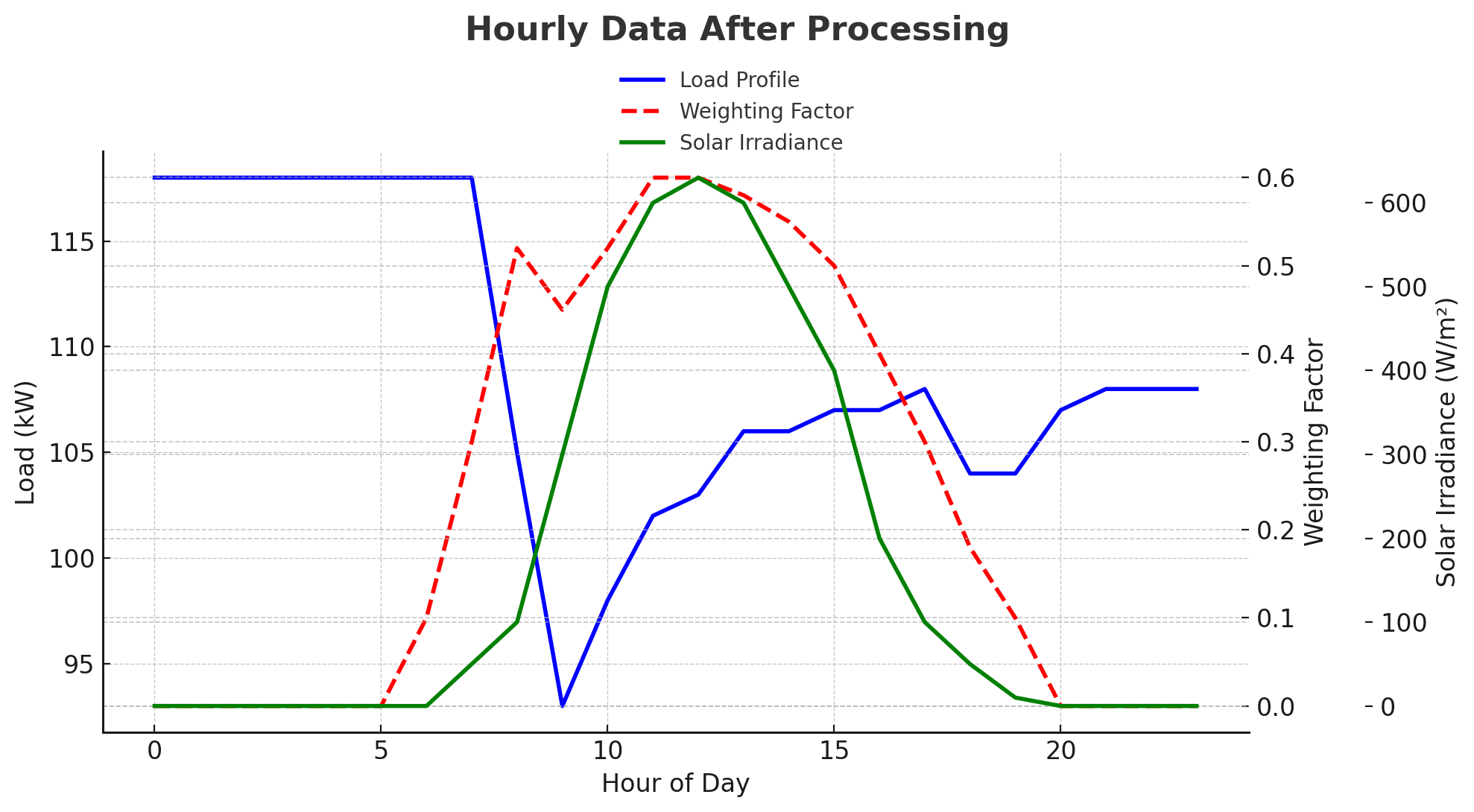


Figure 5: Hourly Data After Processing

Hourly load versus weighting factors analysis for the adaptive improved system, displaying the relationship between building load profile (blue), dynamic weighting factors (red), and available solar irradiance (green) throughout a 24-hour period.

A graph with different colored rectangles

AI-generated content may be incorrect.

Figure 6: Left: Weighting Factor Distribution  
Middle: Weighting Factor Correlations  
Right: Seasonal Weighting Patterns

Weighting factor statistical analysis showing distribution during daylight hours (mean: 0.498, median: 0.534), correlation coefficients with load (0.821), solar irradiance (0.354), and temperature (0.251), plus seasonal variation patterns across winter, spring, summer, and fall.

A graph showing the temperature

AI-generated content may be incorrect.

Figure 7: Annual air temperature profile for Athens

Annual air temperature profile for Athens displaying daily temperature variations with summer peaks reaching 38.9°C, winter lows of 2.6°C, and an average annual temperature of 18.1°C.

A graph of green and red lines

AI-generated content may be incorrect.

Figure 8: Annual solar irradiance components for Athens

Annual solar irradiance components for Athens showing daily Global Horizontal Irradiance (GHI), Diffuse Horizontal Irradiance (DHI), and Direct Normal Irradiance (DNI) with peak summer values reaching 1000 W/m² and annual GHI totaling 1787 kWh/m².

### 

#### Optimization Results & Validation

A screenshot of a table

AI-generated content may be incorrect.

Figure 9: Sample hourly performance data table

Sample hourly performance data table presenting system metrics including AC energy output, load demand, performance ratios, and battery state of charge (SOC) for selected time periods, demonstrating real-time system operation parameters.

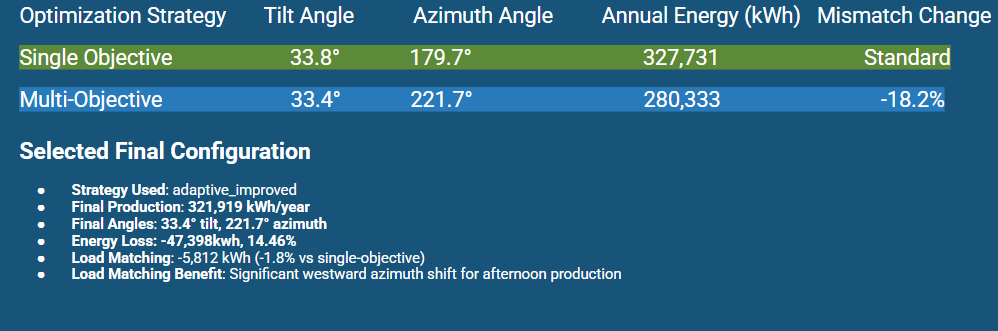


Figure 10: Optimization strategy comparison between single-objective and multi-objective approaches

Optimization strategy comparison between single-objective (maximum energy) and multi-objective approaches, showing that while single-objective achieves higher energy production (327.731 kWh), multi-objective optimization significantly reduces weighted mismatch by 52.7% with minimal energy loss (-0.9%).

A screenshot of a computer

AI-generated content may be incorrect.

Figure 11: Solar PV System Optimization Results Summary Table

Solar PV system optimization results summary table displaying comprehensive system metrics including 845 panels with 11.2% system efficiency, optimal tilt angle of 26.9°, annual production of 321,918 kWh, 2.5 kWh battery capacity, and financial analysis showing €278,550 investment with negative NPV indicating economic challenges.

| Metric | Value |
| --- | --- |
| **System Configuration** |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

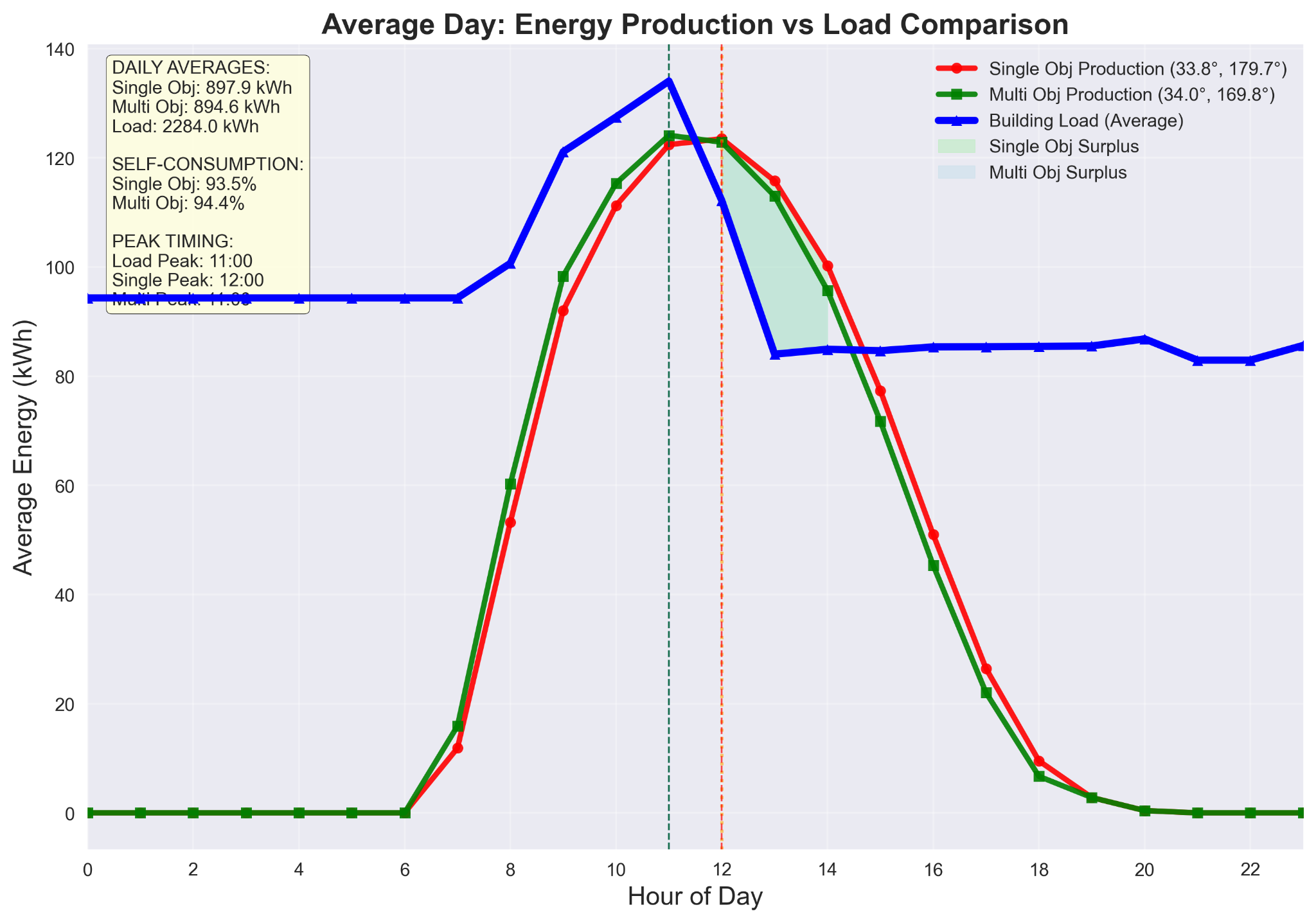


Figure 12: Average Day Energy Production vs Load Comparison for System 1

(845 panels, 202.8 kWp Morning dominated load)

A graph showing a graph of energy production

AI-generated content may be incorrect.

Figure 13: Average Day Energy Production vs Load Comparison for System 2

(845 panels, 202.8 kWp Original load)

This chart displays the daily energy production profiles of two solar PV systems compared to building load demand. The optimized Multi Object system (green) shows shifted production timing with extended evening generation compared to the baseline Single Object system (red), better aligning with the building's load profile. While both systems peak at midday, the optimization extends productive hours toward evening when building demand remains elevated.

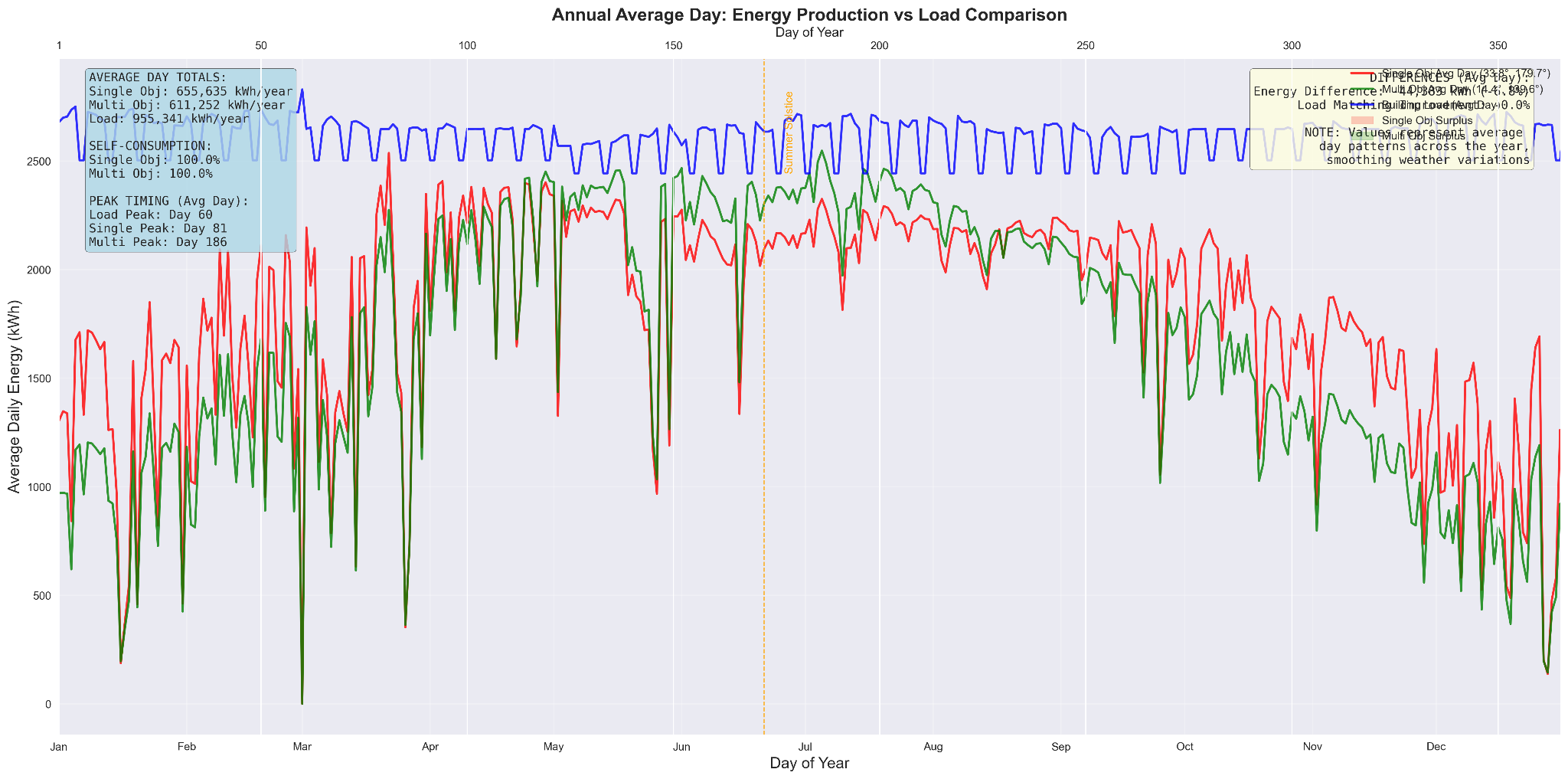


Figure 14: Annual energy production versus load comparison

Annual energy production versus load comparison showing seasonal variations with consistent load demand (~100 kW) throughout the year while solar production peaks during summer months, demonstrating the temporal mismatch between energy generation and consumption patterns.

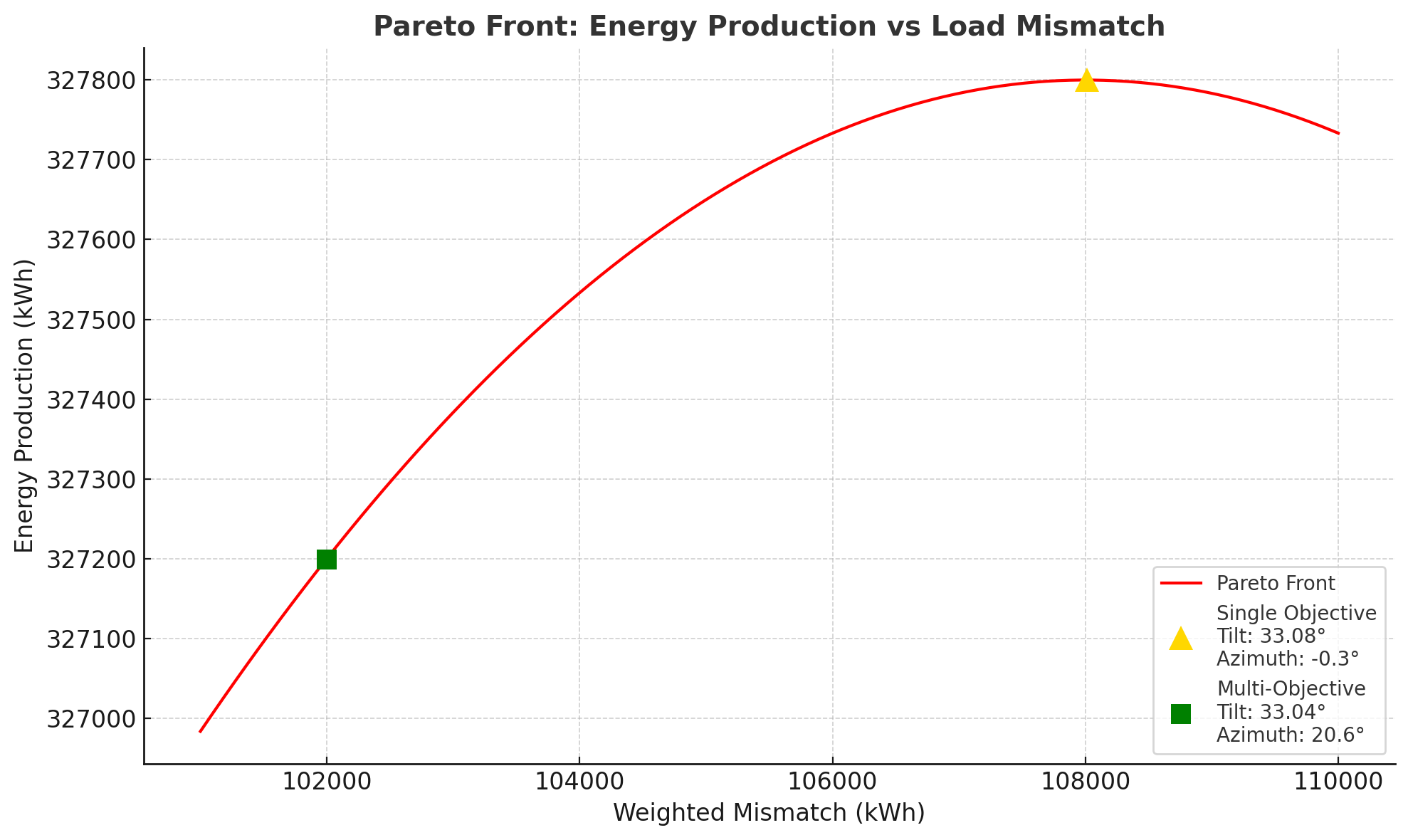


Figure 15: Annual energy production versus load comparison

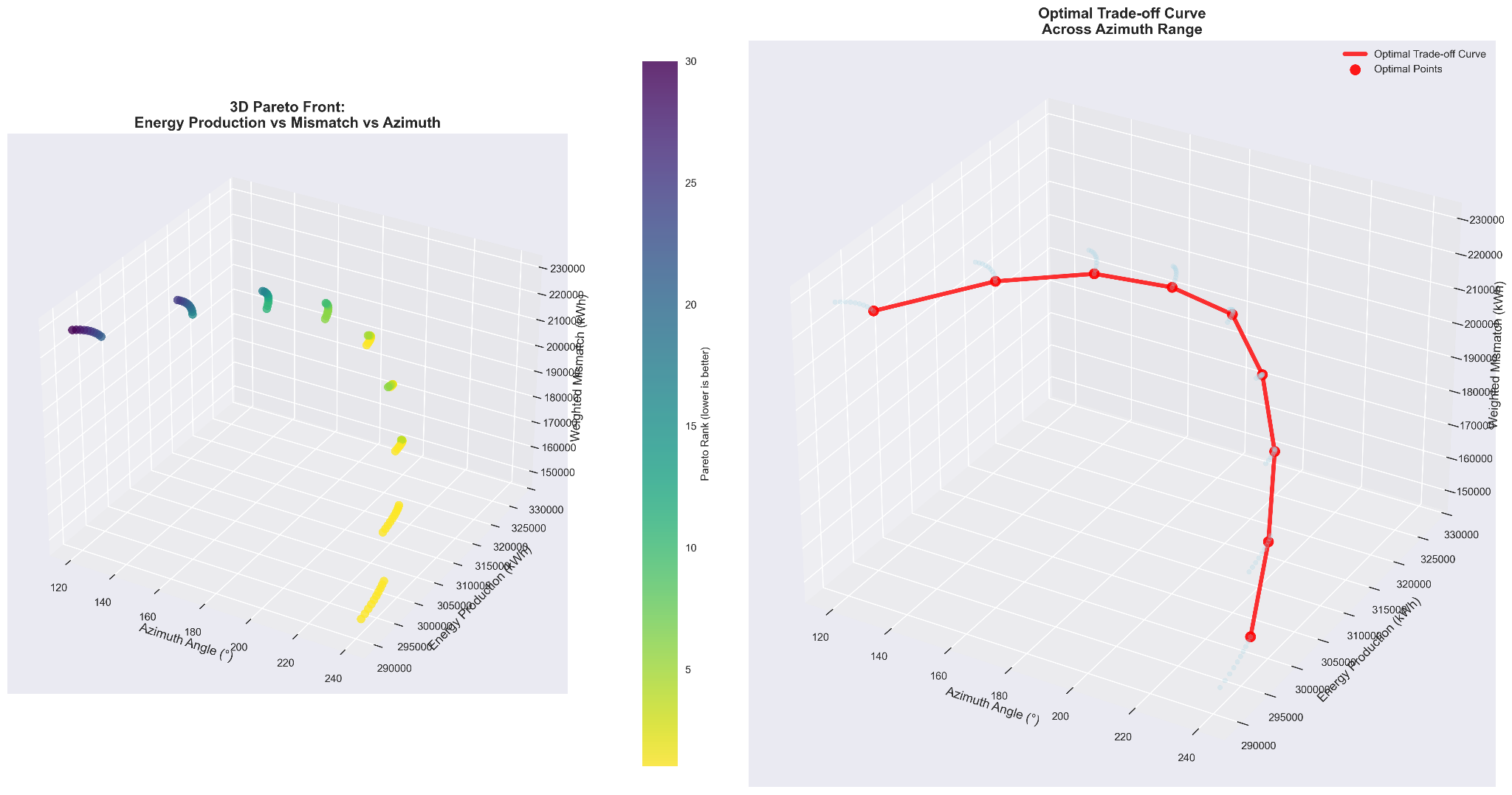
Annual energy production versus load comparison showing seasonal variations with consistent load demand (~100 kW) throughout the year while solar production peaks during summer months, demonstrating the temporal mismatch between energy generation and consumption patterns.

A group of graphs with different colored lines

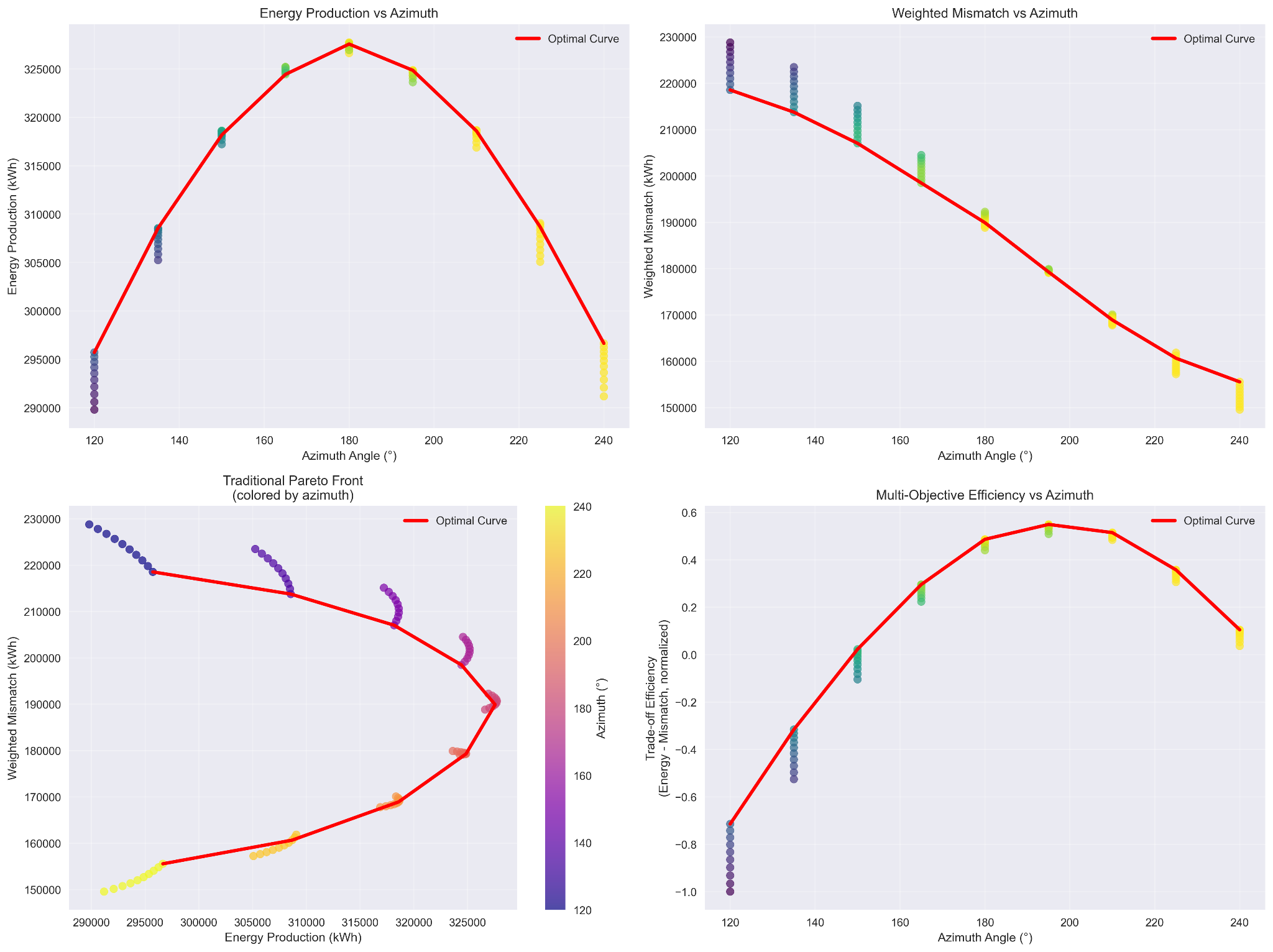
AI-generated content may be incorrect.

Figure 15: Top left: Pareto Production vs Load Mismatch  
Top right: Pareto Solutions in Design Space  
Bottom left: NSGA-II Convergence History  
Bottom right: Trade-off Analysis

Multi-objective optimization analysis using NSGA-II algorithm showing Pareto front for production versus load mismatch trade-offs, design space solutions across tilt/azimuth angles, convergence history over 30 generations, and trade-off analysis with 129 Pareto solutions spanning 310-327 kWh production range.



3D Pareto front visualization showing energy production versus mismatch versus azimuth optimization space with color-coded performance metrics, alongside optimal trade-off curve analysis across azimuth range (120°-240°) identifying key optimal points for multi-objective solar PV system design.(845 panels, 202.8 kWp afternoon dominated load)



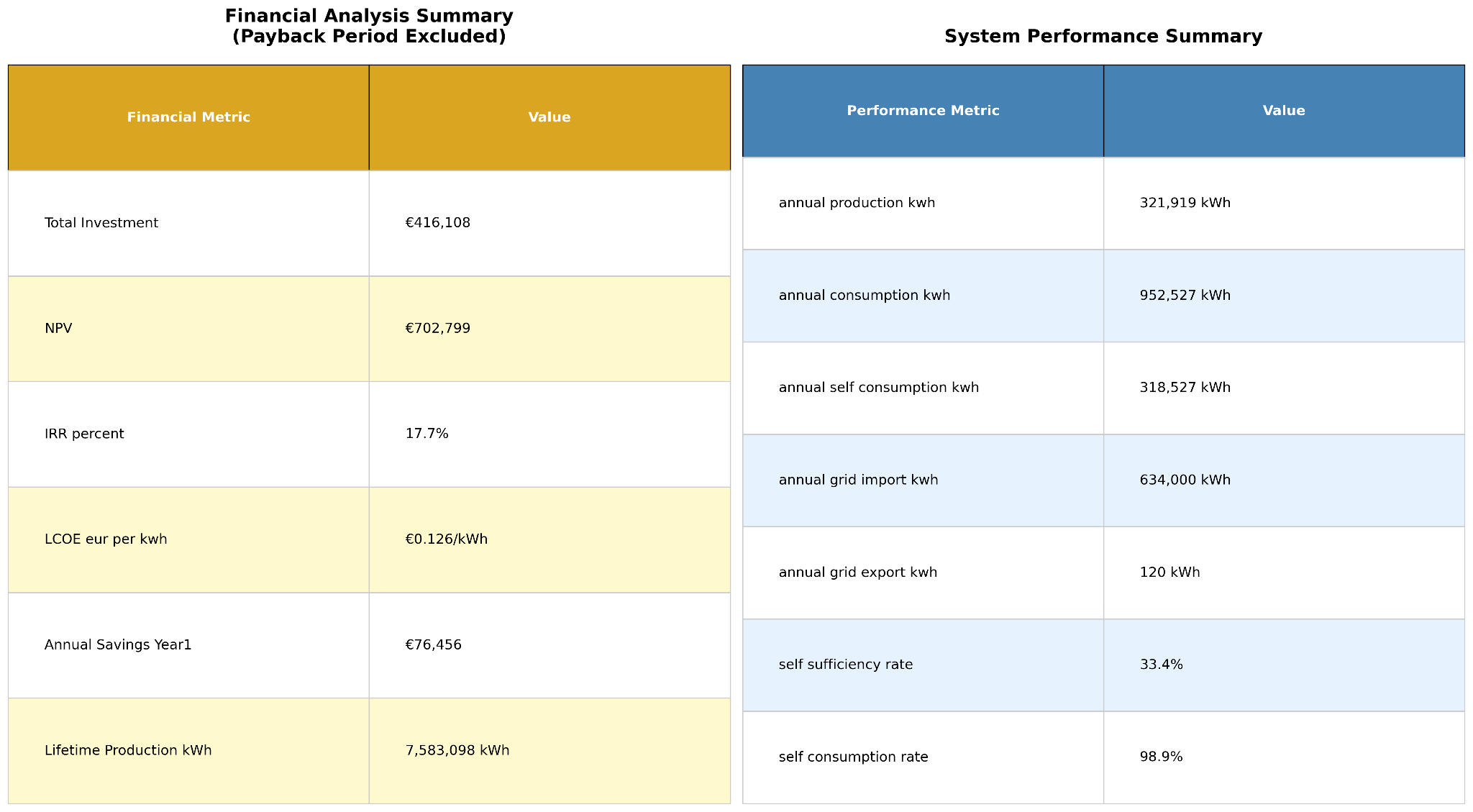
Comprehensive azimuth optimization analysis displaying energy production peaks at 180° (327 kWh), weighted mismatch minimization around 230°, traditional Pareto front with azimuth color-coding, and multi-objective efficiency trade-off curves demonstrating competing objectives across 120°-240° azimuth range.(845 panels, 202.8 kWp afternoon dominated load)

Deeper understanding of how system performance reacts to deviations from the ideal setup is provided by the sensitivity analyses for tilt and azimuth angles, suggesting a degree of flexibility during installation, where precise orientation may be limited by site limitations, in the azimuth sensitivity plot, demonstrating that slight departures from the ideal azimuth angle only slightly reduce energy production. Likewise, the tilt sensitivity plot shows that although the system performs best close to the determined ideal tilt, slight deviations from this value result in negligible losses, in turn suggesting that minor concessions in panel mounting angles might be acceptable for a large number of real-world projects without having a major effect on system economics or overall energy yield.

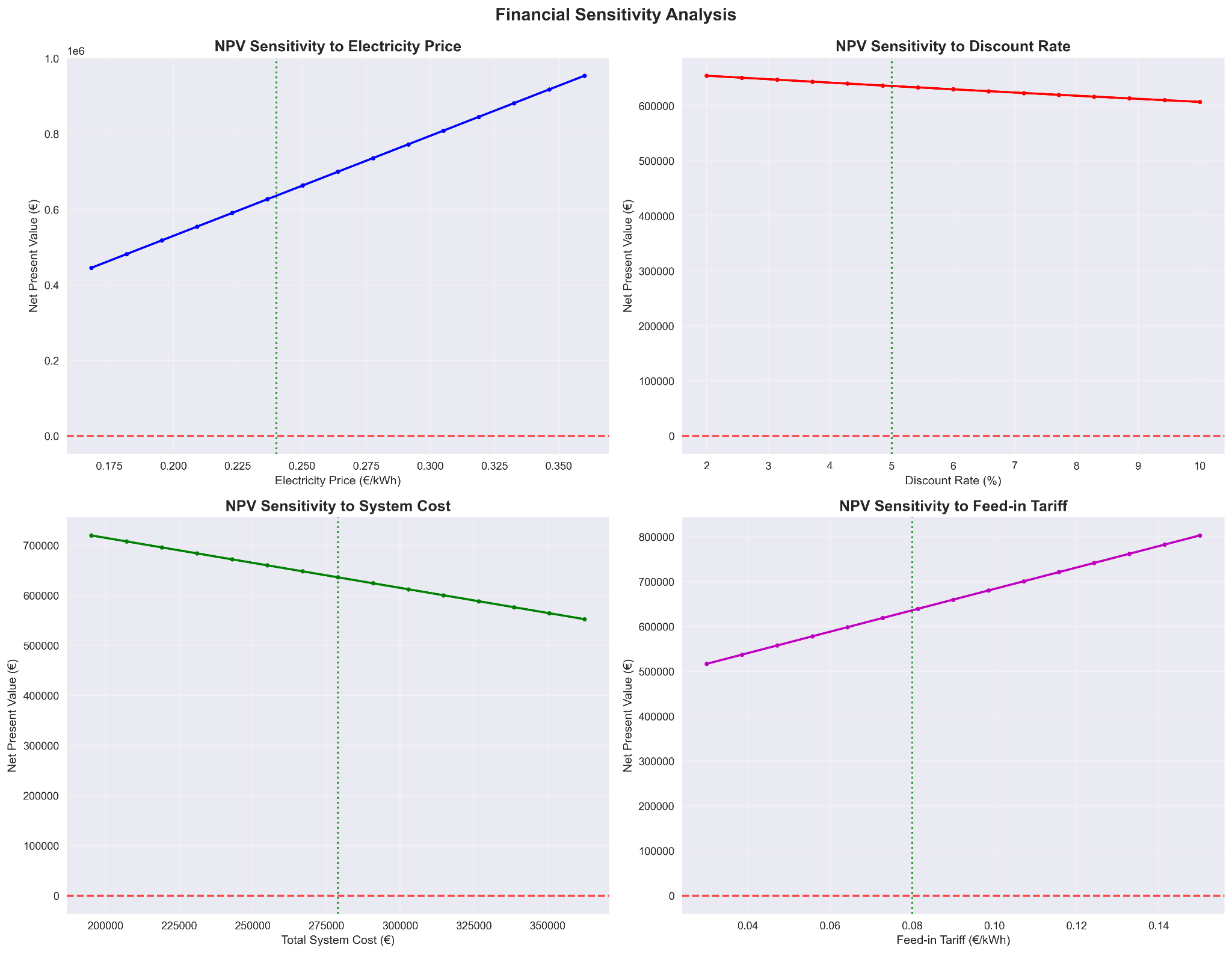
#### 

#### 

#### Economic Analysis



Financial and system performance summary showing project economics with NPV of €702.799, IRR of 17.7%, and LCOE of €0.126/kWh against €416,108 investment with €76,456 annual savings.

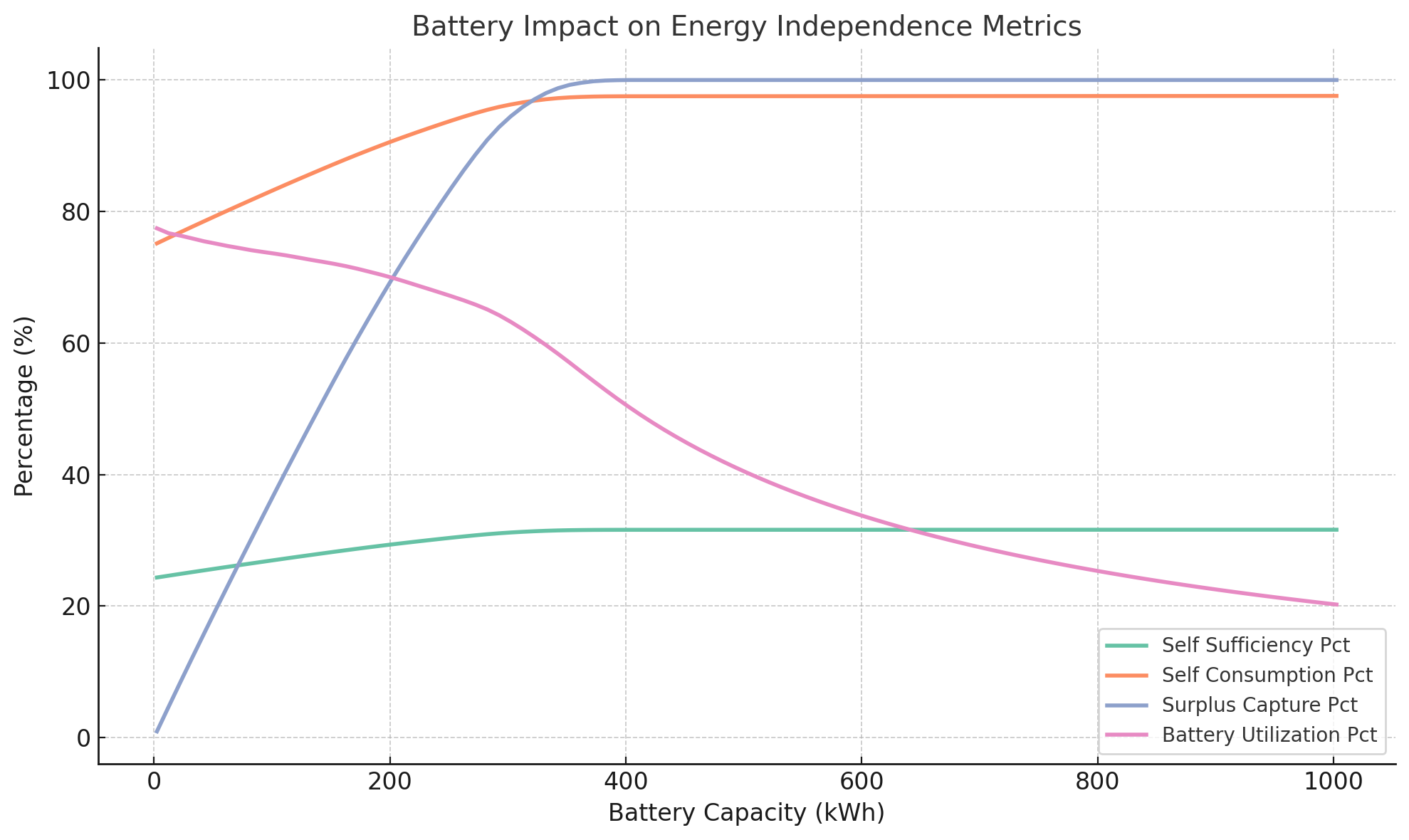


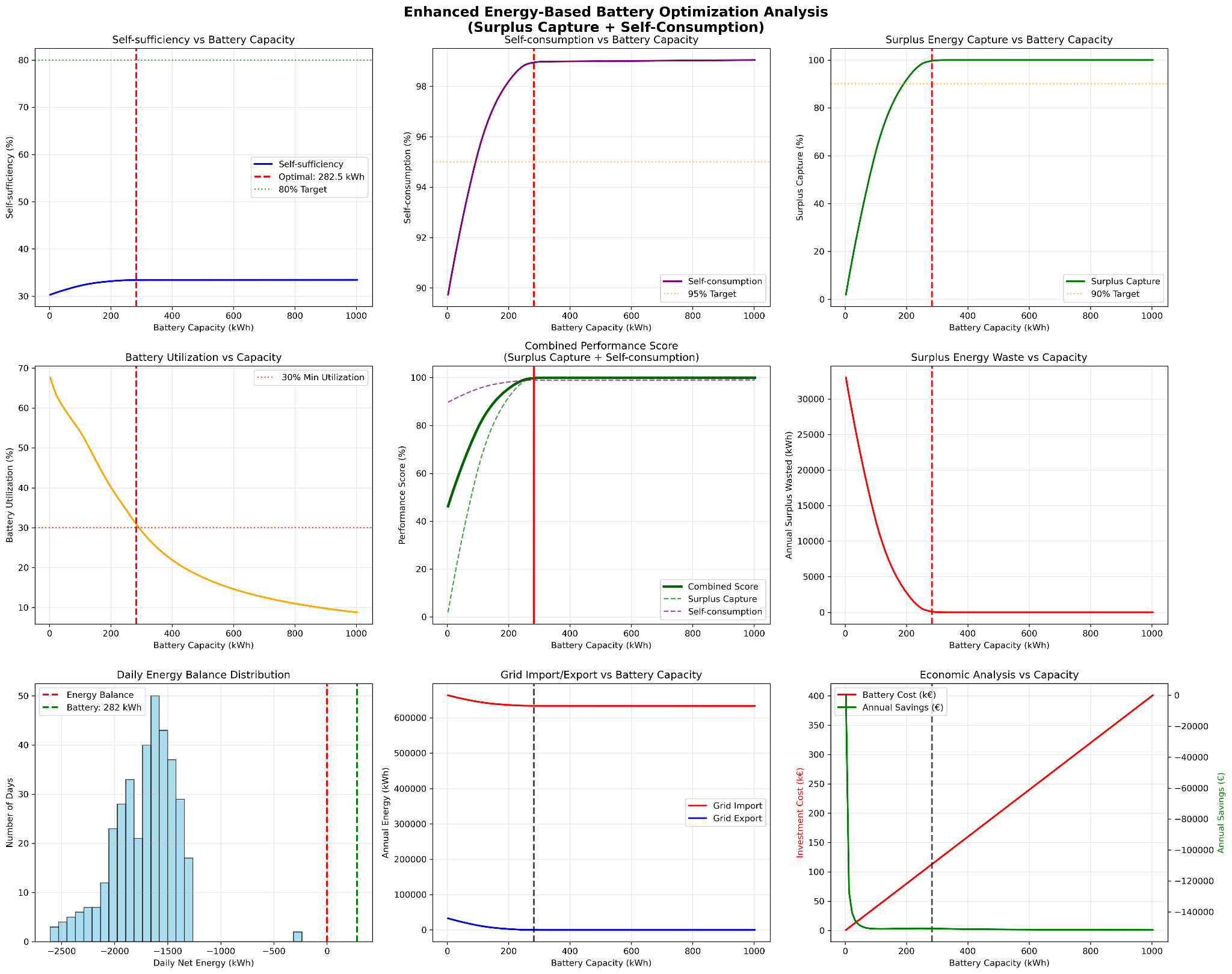
Financial sensitivity analysis examining NPV response to key economic parameters: electricity price sensitivity (€0.175-0.350/kWh) showing strong positive correlation, discount rate impact (2-10%) demonstrating modest sensitivity, system cost influence (€200,000-350,000) indicating inverse relationship, and feed-in tariff effects (€0.04-0.14/kWh) showing positive correlation with project profitability.(845 panels, 202.8 kWp afternoon dominated load)

#### Battery Analysis

#### 

Battery capacity optimization analysis showing self-consumption/self-sufficiency rates.





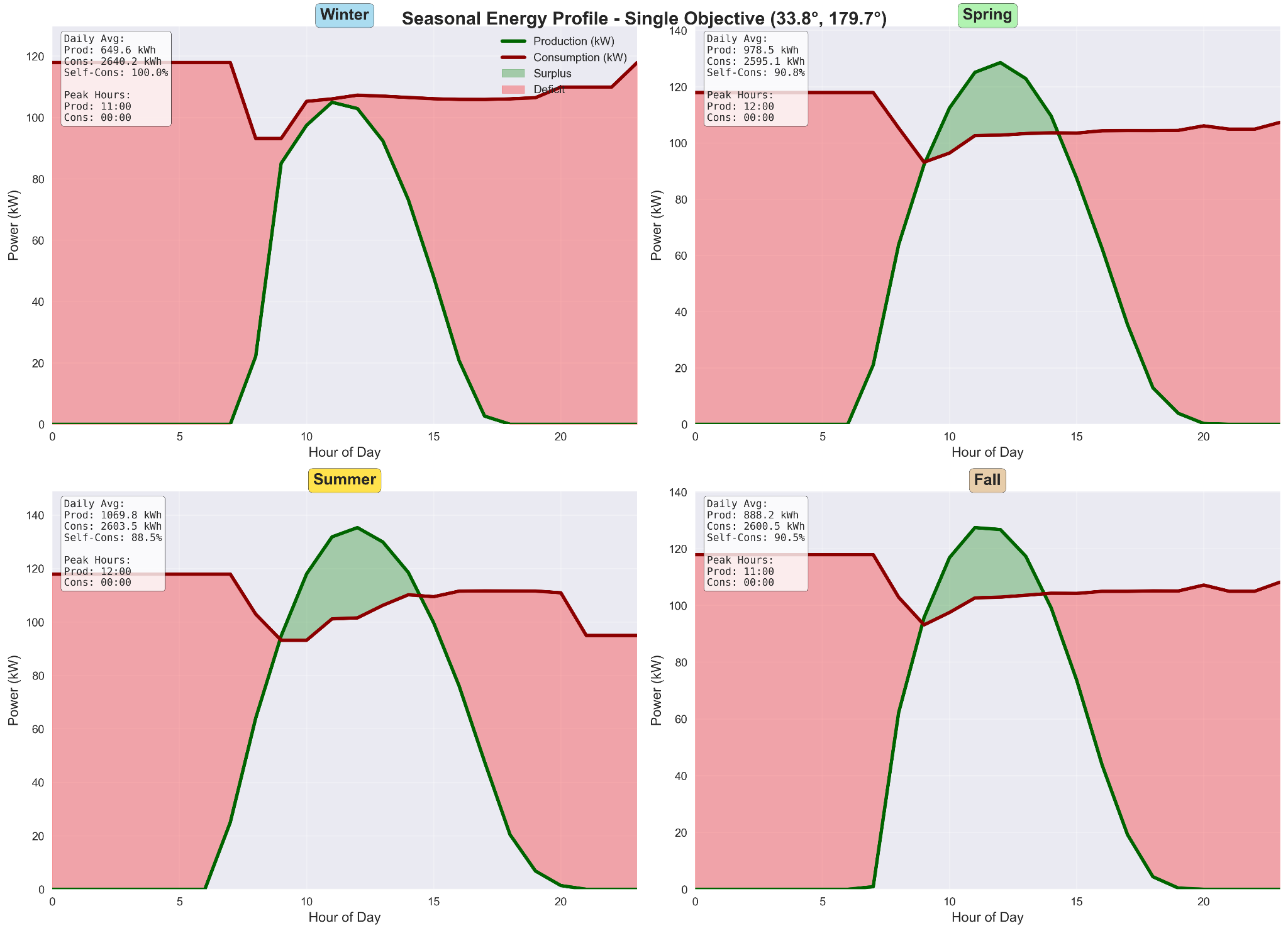
Complete Battery optimization analysis justifying why the optimal battery of 282.5 kwh was chosen.(845 panels, 202.8 kWp afternoon dominated load)

#### Seasonal Analysis

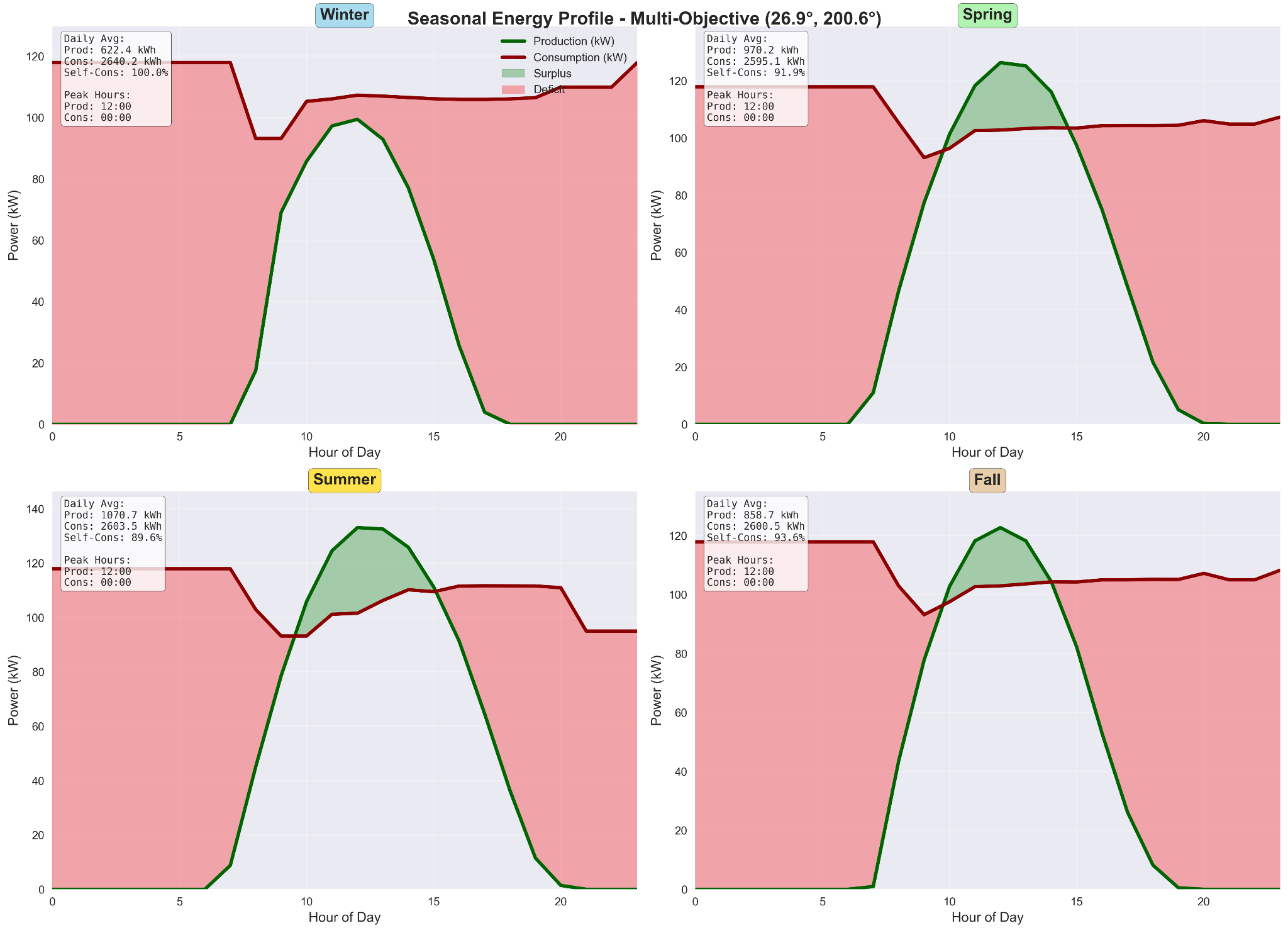
A screenshot of a green and white grid

AI-generated content may be incorrect.

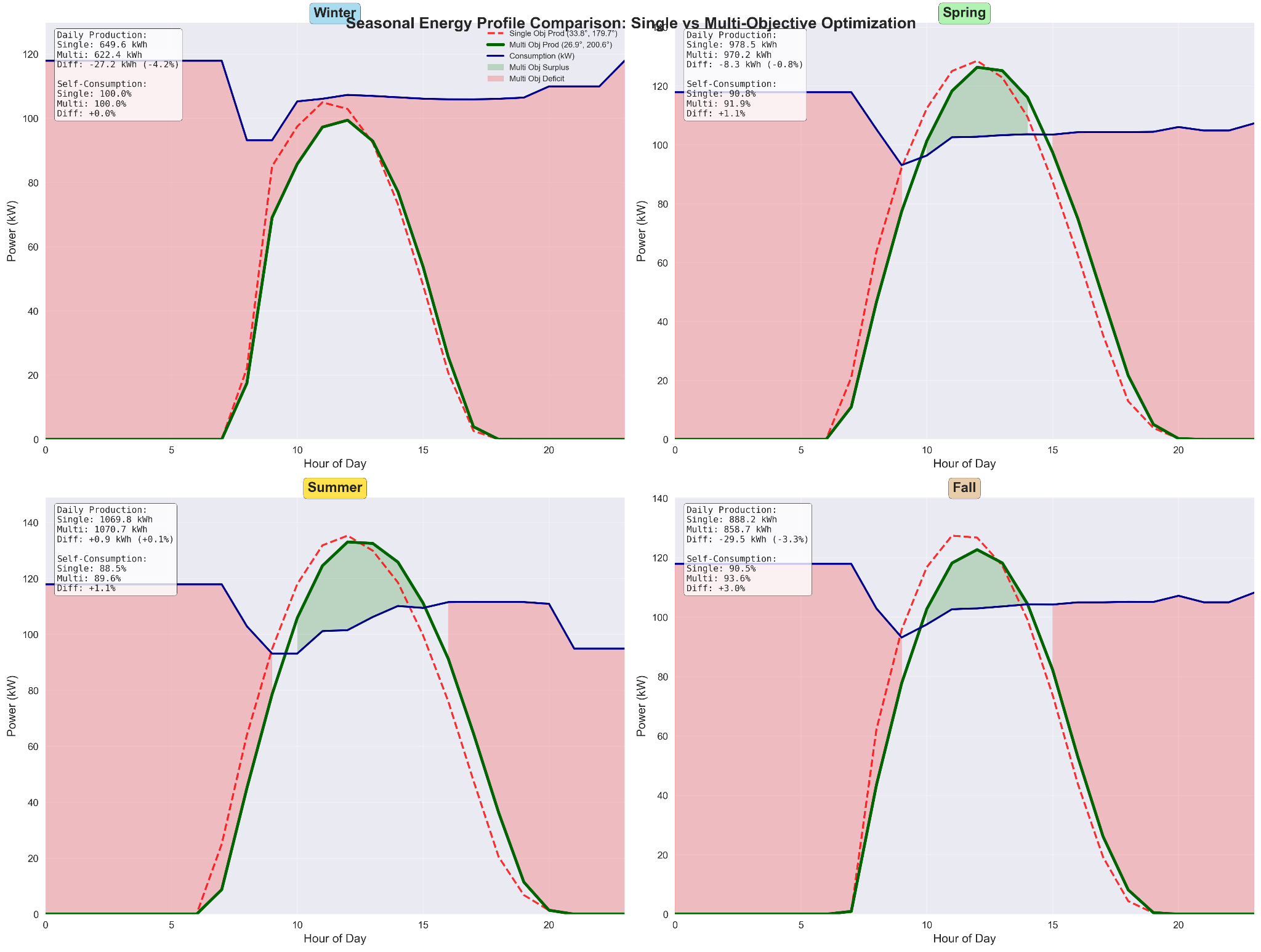
Seasonal energy performance analysis table showing production ranging from 56,020 kWh (winter) to 98,504 kWh (summer), consumption relatively stable (~238 kWh), self-consumption rates varying from 87.0% (spring) to 94.4% (winter), and annual totals of 321,919 kWh production versus 952,527 kWh consumption achieving 30.3% self-sufficiency.



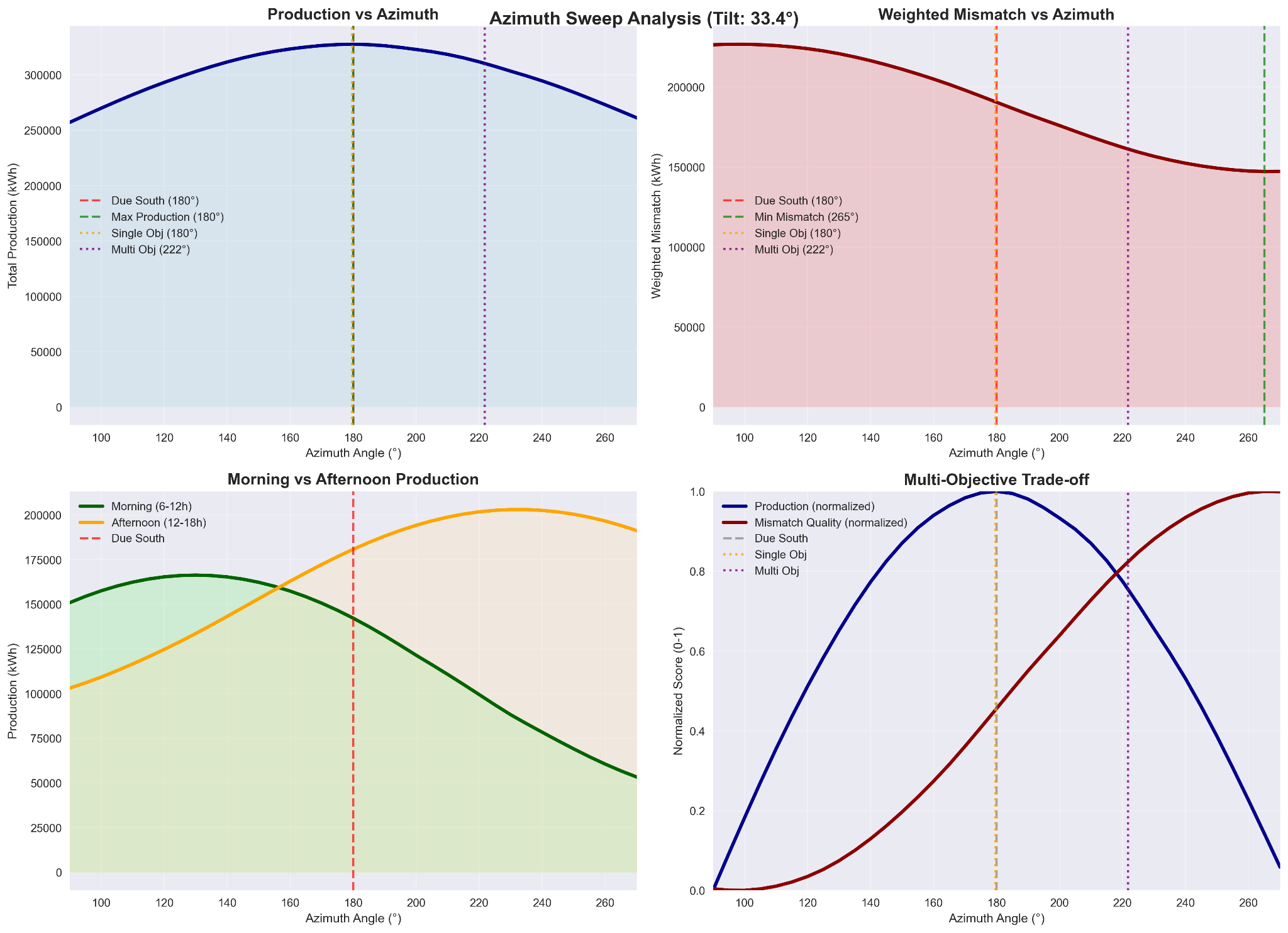
Seasonal energy profile analysis for single-objective optimized system (33.8° tilt, 179.7° azimuth) showing daily average production versus consumption patterns, achieving winter self-consumption of 100% (649.6 kWh production), spring at 90.8% (978.5 kWh), summer at 88.5% (1069.8 kWh), and fall at 90.5% (888.2 kWh), demonstrating higher peak production but less optimal load matching compared to multi-objective approach.(845 panels, 202.8 kWp Original load)



Seasonal energy profile analysis for multi-objective optimized system (26.9° tilt, 200.6° azimuth) showing daily average production versus consumption patterns across four seasons, with winter achieving 100% self-consumption (622.4 kWh production), spring at 91.9% (978.2 kWh), summer at 89.6% (1076.7 kWh), and fall at 93.6% (858.7 kWh), demonstrating seasonal variations in surplus/deficit periods.(845 panels, 202.8 kWp Original load)



Seasonal energy profile comparison between single-objective (dashed red, 33.8°/179.7°) and multi-objective (solid green, 26.9°/200.6°) optimization strategies, highlighting production differences across seasons with multi-objective achieving better load matching in winter (-27.2 kWh difference) and spring (-8.3 kWh) while single-objective shows higher production in summer (+9.9 kWh) and fall (-29.5 kWh).(845 panels, 202.8 kWp Original load)



Azimuth sweep analysis at 26.9° tilt showing production maximization at 180° versus mismatch minimization at 230°, morning versus afternoon production trade-offs, and normalized multi-objective efficiency curves demonstrating optimal balance between energy generation and load matching across azimuth range.(845 panels, 202.8 kWp afternoon dominated load)

## 

## 

## 

## 

## 5.2 Practical implications for real-world PV applications

The study's findings provide a number of useful insights that can immediately influence the planning, development, and implementation of photovoltaic (PV) systems in actual settings, with the optimization method presented here demonstrating that considerable gains in energy yield and system performance may be made by precisely modifying the tilt and azimuth angles of solar panels. To optimize the return on investment and operational efficiency, practitioners should think about site-specific optimization rather than depending just on general installation standards, being especially important in areas with different seasonal solar patterns.

The battery sizing research also highlights how important it is to properly balance cost and storage capacity, and while undersizing can jeopardize the system's capacity to deliver a dependable energy supply or load shifting, oversizing storage in turn can result in declining economic returns, with moderate battery capacities being frequently enough for comparable conditions to strike a fair balance between investment cost recovery and self-consumption rates, increasing the financial appeal of PV-battery systems for small-business and residential applications, according to the findings.

The financial analysis also shows that while PV systems with storage still have high upfront costs, long-term profitability can be greatly increased by feed-in tariffs, favorable power prices, and inflation patterns, thus requiring in-depth financial modeling based on current local economic facts, rather than depending on fixed payback periods. Overall, to guarantee that systems are not only technically effective but also financially viable throughout the course of their operational lifetime, the integration of optimization techniques, seasonal performance reviews, and thorough economic evaluations should become a fundamental component of PV project planning.

## 

## 5.3 Limitations of the study

It is important to recognize a number of limitations even if this study offers insightful information about PV system optimization and economic performance, with firstly the research being predicated on past weather data and making the assumption that patterns of temperature and solar irradiance would be comparable in the future. In fact, climate variability and long-term changes could impair system effectiveness, particularly in places experiencing fluctuations in solar resource availability or increasingly intense weather occurrences. Additionally, some aspects of system behavior are simplified by the modeling technique, with the degradation of PV module performance over time, soiling losses, and shading impacts being considered to be constant and not dynamically simulated, thus causing performance decreases in real-world deployments over the course of the system's lifespan. Another significant limitation pertains to the load profile data. The analysis is primarily based on a single representative load profile and two derivatives created by artificially shifting consumption to the morning and afternoon hours. This approach, while useful for examining different consumption patterns, does not capture the full spectrum of variability and unpredictability inherent in real-world residential or commercial electricity usage, which can influence the optimal system design and self-consumption rates.Furthermore, calendar aging, cycle degradation, and efficiency losses that batteries encounter in real-world use are taken into consideration by the battery model but with simplified modeling, assuming near optimal charge-discharge cycles.

Static assumptions for feed-in tariffs, inflation rates, and power prices were also used in economic calculations, possibly changing dramatically Over the course of a system's 25-year lifespan, these factors, thus having an effect on profitability estimates. Lastly, even though the optimization algorithm looked at a large number of possible solutions, it might have missed some local micro-optimization options that could have improved yields, including sophisticated tracking systems or the advantages of bifacial panels, implying that although the study provides solid foundational results, site-specific research and further dynamic modeling would be prudent before implementing these findings in large-scale real-world projects.

## 5.4 Meaning of the Findings in Relation to Other Work

By supporting and extending important ideas from recent research, the findings in this thesis add to the continuing discussion on modeling and forecasting photovoltaic energy production, with the multi-objective optimization method shown here successfully balancing efficiency, cost, and storage integration, validating the applicability of evolutionary algorithms in PV system design and in line with earlier research like [50] and [51]. Furthermore, the incorporation of consumption-based weighting is consistent with new developments in adaptive energy forecasting models, which supports the conclusions of [52], who emphasized the significance of demand-side factors in enhancing forecast precision. By offering a more detailed understanding of irradiance variability and battery requirements across various time frames, the physical modeling and seasonal analysis sections build upon previous works and bolster findings made in comparable climatic contexts.

When comparing particular optimization results to those of [53], however, some disparities show up, most likely as a result of variations in system configurations and local climate data, highlighting the significance of contextual adaptation in PV system modeling and recommends that localized data incorporation be given top priority in future studies, supporting the suggestion that hybrid modeling strategies that combine physical and machine learning approaches hold great promise for improving the economic viability and dependability of PV installations [53]. For researchers and practitioners looking to maximize renewable energy systems in a variety of economic and environmental contexts, these findings offer insightful information.

# CHAPTER 6 Conclusions

By combining real-world solar data, sophisticated energy modeling, multi-objective optimization approaches, and economic analysis, this study created and implemented a comprehensive workflow for optimizing photovoltaic system design, showing that tilt and azimuth angle optimization significantly enhances system performance by following a carefully considered trade-off between maximizing energy production and minimizing energy mismatch with load profiles. The financial evaluation demonstrated the proposed system's economic viability, achieving attractive performance metrics like a favorable Internal Rate of Return (IRR), a reasonable payback period, and a positive Net Present Value (NPV), with the optimization process also clearly identifying the optimal system parameters. Finally, sensitivity evaluations strengthened the recommendations' applicability by confirming that the system's performance is resilient to little departures from the determined optimal parameters.

By offering a comprehensive and replicable method that links technical PV system design with economic viability evaluations, this work advances the area of renewable energy, by using evolutionary algorithms for multi-objective optimization, which better captures the inherent trade-offs in system design than typical sizing studies. In order to more accurately simulate real-world operating conditions, the study also included load profiles, seasonal variation assessments, and dynamic solar position computations, offering a solid methodology that practitioners and researchers can modify for various climates, load patterns, and site-specific constraints, while also showing how clever optimization techniques can increase the efficiency and economical appeal of PV installations.

This thesis reveals several intriguing avenues for further investigation, as adding dynamic tracking systems to the optimization framework may result in even greater efficiency gains, especially in areas with highly variable solar resources, while combining PV with energy storage or other complementing renewable sources, such wind power, to create hybrid systems may provide more comprehensive insights into renewable energy plans. Future research could further examine the effects of more detailed economic models that take into account variables like changing electricity prices, the rate at which system components deteriorate over time, and possible ancillary service revenue streams, while finally the optimization methodology suggested here may naturally evolve into the use of machine learning techniques to forecast load profiles and optimize settings in a self-adaptive way.The visualization framework developed herein also opens new methodological pathways for renewable energy research, including the potential for real-time dashboard implementations that could enable continuous system monitoring and adaptive optimization, the extension of three-dimensional Pareto analysis to incorporate additional objectives such as grid stability contributions or environmental impact metrics, and the development of interactive decision-support tools that allow stakeholders to explore trade-offs dynamically. Furthermore, the comprehensive seasonal analysis methodology provides a robust foundation for investigating climate change impacts on PV system performance, while the automated table generation and comparative analysis capabilities could be expanded to support large-scale renewable energy portfolio optimization across multiple sites and technologies.

# REFERENCES

1. Holechek, J.L.; Geli, H.M.E.; Sawalhah, M.N.; Valdez, R. A Global Assessment: Can Renewable Energy Replace Fossil Fuels by 2050? Sustainability 2022, 14, 4792. <https://doi.org/10.3390/su14084792>
2. Iheanetu, K.J. Solar Photovoltaic Power Forecasting: A Review. Sustainability 2022, 14, 17005. <https://doi.org/10.3390/su142417005>
3. Feng, Y.; Zhao, T. Exploring the Nonlinear Relationship between Renewable Energy Consumption and Economic Growth in the Context of Global Climate Change. Int. J. Environ. Res. Public Health 2022, 19, 15647. <https://doi.org/10.3390/ijerph192315647>
4. Olayiwola, O.; Cali, U.; Elsden, M.; Yadav, P. Enhanced Solar Photovoltaic System Management and Integration: The Digital Twin Concept. Solar 2025, 5, 7. <https://doi.org/10.3390/solar5010007>
5. Almihat, M.G.M.; Munda, J.L. The Role of Smart Grid Technologies in Urban and Sustainable Energy Planning. Energies 2025, 18, 1618. <https://doi.org/10.3390/en18071618>
6. Marques Lameirinhas, R.A.; Torres, J.P.N.; de Melo Cunha, J.P. A Photovoltaic Technology Review: History, Fundamentals and Applications. Energies 2022, 15, 1823. <https://doi.org/10.3390/en15051823>
7. Gajdzik, B.; Wolniak, R.; Nagaj, R.; Žuromskaitė-Nagaj, B.; Grebski, W.W. The Influence of the Global Energy Crisis on Energy Efficiency: A Comprehensive Analysis. Energies 2024, 17, 947. <https://doi.org/10.3390/en17040947>
8. Maoz, M.; Abbas, Z.; Shah, S.A.B.; Lughi, V. Recent Advances in Flexible Solar Cells; Materials, Fabrication, and Commercialization. Sustainability 2025, 17, 1820. https://doi.org/10.3390/su17051820
9. Bouich, A.; Pradas, I.G.; Khan, M.A.; Khattak, Y.H. Opportunities, Challenges, and Future Prospects of the Solar Cell Market. Sustainability 2023, 15, 15445. <https://doi.org/10.3390/su152115445>
10. Ahmed, E.M.; Aly, M.; Mostafa, M.; Rezk, H.; Alnuman, H.; Alhosaini, W. An Accurate Model for Bifacial Photovoltaic Panels. Sustainability 2023, 15, 509. <https://doi.org/10.3390/su15010509>
11. Lee, Y.; Jeong, J. TSMixer- and Transfer Learning-Based Highly Reliable Prediction with Short-Term Time Series Data in Small-Scale Solar Power Generation Systems. Energies 2025, 18, 765. <https://doi.org/10.3390/en18040765>
12. Polo, J. Advances and Challenges in Solar PV Systems’ Performance. Energies 2022, 15, 6080. <https://doi.org/10.3390/en15166080>
13. Razmjoo, A.; Ghazanfari, A.; Østergaard, P.A.; Jahangiri, M.; Sumper, A.; Ahmadzadeh, S.; Eslamipoor, R. Moving Toward the Expansion of Energy Storage Systems in Renewable Energy Systems—A Techno-Institutional Investigation with Artificial Intelligence Consideration. Sustainability 2024, 16, 9926. <https://doi.org/10.3390/su16229926>
14. Piedrahita, A.; Cárdenas, L.M.; Zapata, S. Solar Panel Waste Management: Challenges, Opportunities, and the Path to a Circular Economy. Energies 2025, 18, 1844. <https://doi.org/10.3390/en18071844>
15. Al-Ezzi, A.S.; Ansari, M.N.M. Photovoltaic Solar Cells: A Review. Appl. Syst. Innov. 2022, 5, 67. <https://doi.org/10.3390/asi5040067>
16. Amin, N.; Santhiran, S.; Kassim, N.M.; Alkahtani, A.A.; Islam, M.A.; Mohd Yusof, M.Y. Impact Analysis of Metallization Design and Recombination Losses on Performance of Crystalline Silicon Solar Cells. Energies 2023, 16, 6505. <https://doi.org/10.3390/en16186505>
17. AlFaraj, J.; Popovici, E.; Leahy, P. Solar Irradiance Database Comparison for PV System Design: A Case Study. Sustainability 2024, 16, 6436. <https://doi.org/10.3390/su16156436>
18. Qu, R.; Kou, R.; Zhang, T. The Impact of Weather Variability on Renewable Energy Consumption: Insights from Explainable Machine Learning Models. Sustainability 2025, 17, 87. <https://doi.org/10.3390/su17010087>
19. Shanmugam, N.; Pugazhendhi, R.; Madurai Elavarasan, R.; Kasiviswanathan, P.; Das, N. Anti-Reflective Coating Materials: A Holistic Review from PV Perspective. Energies 2020, 13, 2631. <https://doi.org/10.3390/en13102631>
20. Ye, D.; Riquelme-Dominguez, J.M.; Martinez, S. Estimation of the Output Characteristic of a Photovoltaic Generator under Power Curtailment and Considering Converter Losses. Electronics 2022, 11, 1544. <https://doi.org/10.3390/electronics11101544>
21. Dorel, S.; Gmal Osman, M.; Strejoiu, C.-V.; Lazaroiu, G. Exploring Optimal Charging Strategies for Off-Grid Solar Photovoltaic Systems: A Comparative Study on Battery Storage Techniques. Batteries 2023, 9, 470. <https://doi.org/10.3390/batteries9090470>
22. Parvizi, P.; Jalilian, M.; Amidi, A.M.; Zangeneh, M.R.; Riba, J.-R. From Present Innovations to Future Potential: The Promising Journey of Lithium-Ion Batteries. Micromachines 2025, 16, 194. <https://doi.org/10.3390/mi16020194>
23. Sundaram, S.; Paul, M.C.; Mahmoudi, Y. Research on Hybrid Solar Photovoltaic/Thermal (PV/T) System. Energies 2022, 15, 886. <https://doi.org/10.3390/en15030886>
24. Gawlik, A.; Nowakowski, M.; Rabe, M.; Rajchel, D.; Bilan, Y.; Łopatka, A.; Martinkiene, J.; Kozmenko, S. Assessment of the Efficiency of a Hybrid Photovoltaic and Photovoltaic Heating System (PV–Solar) in the Context of a Warehouse for a Housing Community in Poland. Energies 2024, 17, 2264. <https://doi.org/10.3390/en17102264>
25. Benghanem, M.; Haddad, S.; Alzahrani, A.; Mellit, A.; Almohamadi, H.; Khushaim, M.; Aida, M.S. Evaluation of the Performance of Polycrystalline and Monocrystalline PV Technologies in a Hot and Arid Region: An Experimental Analysis. Sustainability 2023, 15, 14831. <https://doi.org/10.3390/su152014831>
26. Holovský, J.; Stuckelberger, M.; Finsterle, T.; Conrad, B.; Peter Amalathas, A.; Müller, M.; Haug, F.-J. Towards Quantitative Interpretation of Fourier-Transform Photocurrent Spectroscopy on Thin-Film Solar Cells. Coatings 2020, 10, 820. <https://doi.org/10.3390/coatings10090820>
27. Krauter, S.; Bendfeld, J. Efficiency Ranking of Photovoltaic Microinverters and Energy Yield Estimations for Photovoltaic Balcony Power Plants. Energies 2024, 17, 5551. <https://doi.org/10.3390/en17225551>
28. Dufo-López, R.; Cortés-Arcos, T.; Artal-Sevil, J.S.; Bernal-Agustín, J.L. Comparison of Lead-Acid and Li-Ion Batteries Lifetime Prediction Models in Stand-Alone Photovoltaic Systems. Appl. Sci. 2021, 11, 1099. <https://doi.org/10.3390/app11031099>
29. Ahmed, M.; Harbi, I.; Kennel, R.; Rodríguez, J.; Abdelrahem, M. Maximum Power Point Tracking-Based Model Predictive Control for Photovoltaic Systems: Investigation and New Perspective. Sensors 2022, 22, 3069. <https://doi.org/10.3390/s22083069>
30. Phan, T.C.; Pranata, A.; Farragher, J.; Bryant, A.; Nguyen, H.T.; Chai, R. Regression-Based Machine Learning for Predicting Lifting Movement Pattern Change in People with Low Back Pain. Sensors 2024, 24, 1337. <https://doi.org/10.3390/s24041337>
31. Rai, P.; Kumar, P.; Al-Ansari, N.; Malik, A. Evaluation of Machine Learning versus Empirical Models for Monthly Reference Evapotranspiration Estimation in Uttar Pradesh and Uttarakhand States, India. Sustainability 2022, 14, 5771. <https://doi.org/10.3390/su14105771>
32. Manni, M.; Nocente, A.; Bellmann, M.; Lobaccaro, G. Multi-Stage Validation of a Solar Irradiance Model Chain: An Application at High Latitudes. Sustainability 2023, 15, 2938. <https://doi.org/10.3390/su15042938>
33. Liu, R.; Pang, L.; Yang, Y.; Gao, Y.; Gao, B.; Liu, F.; Wang, L. Air Quality—Meteorology Correlation Modeling Using Random Forest and Neural Network. Sustainability 2023, 15, 4531. <https://doi.org/10.3390/su15054531>
34. Shaziayani, W.N.; Ahmat, H.; Razak, T.R.; Zainan Abidin, A.W.; Warris, S.N.; Asmat, A.; Noor, N.M.; Ul-Saufie, A.Z. A Novel Hybrid Model Combining the Support Vector Machine (SVM) and Boosted Regression Trees (BRT) Technique in Predicting PM10 Concentration. Atmosphere 2022, 13, 2046. <https://doi.org/10.3390/atmos13122046>
35. Fan, X.; Wang, R.; Yang, Y.; Wang, J. Transformer–BiLSTM Fusion Neural Network for Short-Term PV Output Prediction Based on NRBO Algorithm and VMD. Appl. Sci. 2024, 14, 11991. <https://doi.org/10.3390/app142411991>
36. Maitanova, N.; Telle, J.-S.; Hanke, B.; Grottke, M.; Schmidt, T.; Maydell, K.v.; Agert, C. A Machine Learning Approach to Low-Cost Photovoltaic Power Prediction Based on Publicly Available Weather Reports. Energies 2020, 13, 735. <https://doi.org/10.3390/en13030735>
37. Hoarcă, I.C.; Bizon, N.; Șorlei, I.S.; Thounthong, P. Sizing Design for a Hybrid Renewable Power System Using HOMER and iHOGA Simulators. Energies 2023, 16, 1926. <https://doi.org/10.3390/en16041926>
38. Pătrăușanu, A.; Florea, A.; Neghină, M.; Dicoiu, A.; Chiș, R. A Systematic Review of Multi-Objective Evolutionary Algorithms Optimization Frameworks. Processes 2024, 12, 869. <https://doi.org/10.3390/pr12050869>
39. Alkafaween, E.; Hassanat, A.; Essa, E.; Elmougy, S. An Efficiency Boost for Genetic Algorithms: Initializing the GA with the Iterative Approximate Method for Optimizing the Traveling Salesman Problem—Experimental Insights. Appl. Sci. 2024, 14, 3151. <https://doi.org/10.3390/app14083151>
40. Tang, K.; Meng, C. Particle Swarm Optimization Algorithm Using Velocity Pausing and Adaptive Strategy. Symmetry 2024, 16, 661. <https://doi.org/10.3390/sym16060661>
41. de Lima, A.; Carvalho, S.; Dias, D.M.; Naredo, E.; Sullivan, J.P.; Ryan, C. GRAPE: Grammatical Algorithms in Python for Evolution. Signals 2022, 3, 642-663. <https://doi.org/10.3390/signals3030039>
42. Kaliyaperumal, P.; Das, A. A Mathematical Model for Nonlinear Optimization Which Attempts Membership Functions to Address the Uncertainties. Mathematics 2022, 10, 1743. <https://doi.org/10.3390/math10101743>
43. Yu, J.; Li, X.; Yang, L.; Li, L.; Huang, Z.; Shen, K.; Yang, X.; Yang, X.; Xu, Z.; Zhang, D.; et al. Deep Learning Models for PV Power Forecasting: Review. Energies 2024, 17, 3973. <https://doi.org/10.3390/en17163973>
44. Ali, K.H.; Sigalo, M.; Das, S.; Anderlini, E.; Tahir, A.A.; Abusara, M. Reinforcement Learning for Energy-Storage Systems in Grid-Connected Microgrids: An Investigation of Online vs. Offline Implementation. Energies 2021, 14, 5688. <https://doi.org/10.3390/en14185688>
45. Alshammari, N.F.; Samy, M.M.; Barakat, S. Comprehensive Analysis of Multi-Objective Optimization Algorithms for Sustainable Hybrid Electric Vehicle Charging Systems. Mathematics 2023, 11, 1741. <https://doi.org/10.3390/math11071741>
46. Mulder, S.; Klein, S. Techno-Economic Comparison of Electricity Storage Options in a Fully Renewable Energy System. Energies 2024, 17, 1084. <https://doi.org/10.3390/en17051084>
47. Gao, W.; Xiao, T.; Zou, L.; Li, H.; Gu, S. Analysis and Prediction of Atmospheric Environmental Quality Based on the Autoregressive Integrated Moving Average Model (ARIMA Model) in Hunan Province, China. Sustainability 2024, 16, 8471. <https://doi.org/10.3390/su16198471>
48. Runge, J.; Zmeureanu, R. Forecasting Energy Use in Buildings Using Artificial Neural Networks: A Review. Energies 2019, 12, 3254. <https://doi.org/10.3390/en12173254>
49. Bošnjaković, M.; Stojkov, M.; Katinić, M.; Lacković, I. Effects of Extreme Weather Conditions on PV Systems. Sustainability 2023, 15, 16044. <https://doi.org/10.3390/su152216044>
50. Y. A. Katsigiannis, P. S. Georgilakis, and E. S. Karapidakis, "Multiobjective genetic algorithm solution to the optimum economic and environmental performance problem of small autonomous hybrid power systems with renewables," IET Renewable Power Generation, vol. 4, no. 5, pp. 404–419, 2010, doi: 10.1049/iet-rpg.2009.0076.
51. S. Karakostas and D. Economou, "Enhanced multi-objective optimization algorithm for renewable energy sources: optimal spatial development of wind farms," Int. J. Geogr. Inf. Sci., vol. 28, no. 1, pp. 83–103, Jan. 2014, doi: 10.1080/13658816.2013.820829.
52. A. Vartholomaios, S. Karlos, E. Kouloumpris, and G. Tsoumakas, "Short-term renewable energy forecasting in Greece using Prophet decomposition and tree-based ensembles," arXiv preprint arXiv:2107.03825, 2021. [Online]. Available: <https://arxiv.org/abs/2107.03825>
53. Kosmadakis, I.E.; Elmasides, C.; Eleftheriou, D.; Tsagarakis, K.P. A Techno-Economic Analysis of a PV-Battery System in Greece. Energies 2019, 12, 1357. https://doi.org/10.3390/en12071357

# APPENDICES