ITS8080 Energy Data Science

Samuel Heinrich - 252145MV

December 13, 2025

Abstract

This report presents a comprehensive data science framework for optimizing a Home Energy Management System (HEMS). Addressing the challenges of the "Energy Trilemma," I develop an end-to-end pipeline that integrates data cleaning, advanced feature engineering, time series forecasting, and mathematical optimization. Using Seasonal-Trend decomposition (STL) methods, I analyze hourly consumption and generation data. For forecasting, I compare classical Seasonal ARIMA (SARIMA) models against non-linear Machine Learning approaches (XGBoost). The XGBoost model, enriched with exogenous weather variables and engineered temporal features, demonstrates superior performance in a rigorous walk-forward validation, achieving the lowest Root Mean Squared Error (RMSE). Finally, I leverage these forecasts in a Linear Programming (LP) optimization model to schedule battery storage operations. The results quantify the economic benefits of intelligent energy management, demonstrating significant cost reductions through price arbitrage and maximized self-consumption. For the code itself, visit my Github.

1 Introduction: Digital Transformation of the Energy Sector

1.1 Context: The Energy Trilemma

The global energy landscape faces what policymakers call the "Energy Trilemma"—the challenge of simultaneously achieving energy security, affordability, and environmental sustainability (World Energy Council, 2019). These three goals often conflict: cheap fossil fuels harm the environment, while renewable sources introduce supply variability. The rise of digitalization offers a path forward by making the grid smarter and more responsive (International Energy Agency, 2017; Tuballa & Abundo, 2016). This project zooms in on the residential sector, where Home Energy Management Systems (HEMS) put households in control of their own energy flows (Beaudin & Zareipour, 2015; Zhou et al., 2016).

1.2 Dataset Overview

The analysis uses hourly time series data from a European household, covering one full year (July 2013 to June 2014). The dataset captures the essential variables needed to understand and optimize home energy use:

- **Demand (kW):** How much electricity the household consumes each hour. This depends on occupant behavior—when people cook, watch TV, or run the washing machine.
- **PV Generation (kW):** Electricity produced by rooftop solar panels. Output follows the sun's path and drops to zero at night.
- **Price (/MWh):** The cost of electricity from the grid, which varies by time of day. Prices tend to rise when many households draw power simultaneously.
- **Weather Variables:** Temperature and solar irradiance, which drive both heating/cooling demand and PV output.

1.3 Initial Visualization

Figure 1 shows PV generation, household demand, and electricity price over the first week of July 2013. Several patterns stand out immediately:

- **PV generation** follows a bell curve each day, peaking around noon and falling to zero after sunset.
- **Demand** shows morning and evening spikes when occupants are active at home.
- **Price** fluctuates throughout the day, with peaks often aligning with high-demand periods.

The key observation is a *timing mismatch*: solar panels produce the most power at midday, but households consume the most in the evening. This gap motivates the use of battery storage to shift solar energy from when it is generated to when it is needed.

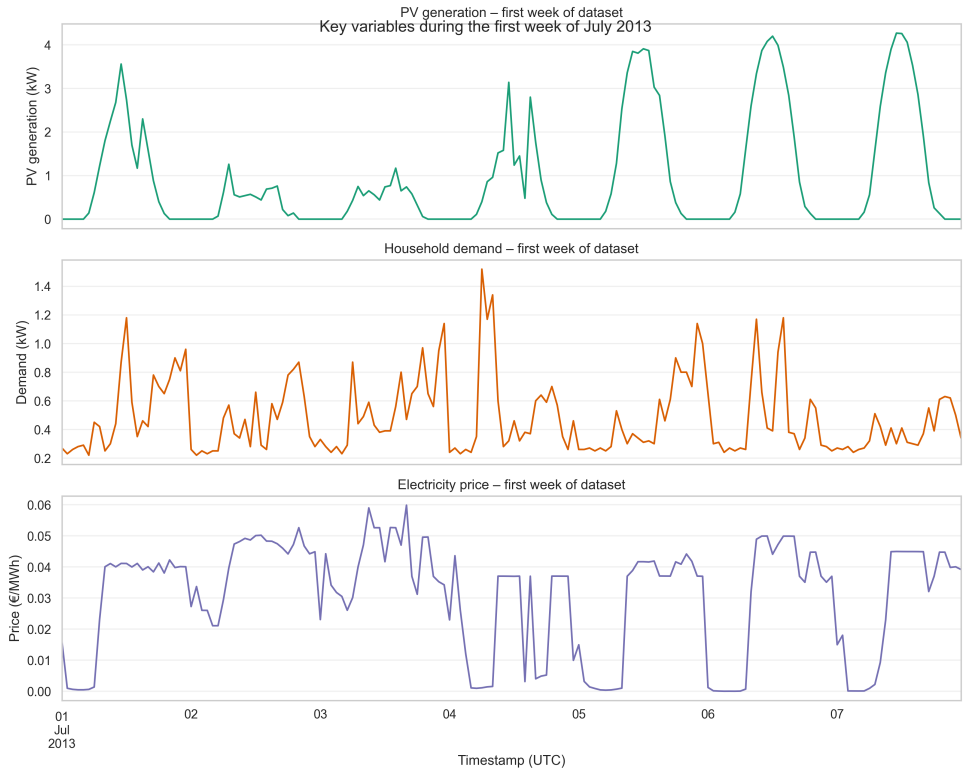

Figure 1: Key variables during the first week of July 2013.

1.4 How Digitalization Transforms Household Energy

The traditional electricity system was one-directional: large power plants generated electricity, the grid delivered it, and households consumed it. Digitalization changes this model in several ways:

- **Smart meters** record consumption at fine time intervals, giving both utilities and consumers visibility into usage patterns (Wang et al., 2019).
- **Connected devices** (thermostats, appliances, EV chargers) can receive signals and adjust their operation based on grid conditions or price signals.
- **Home batteries** store excess solar energy for later use, reducing dependence on the grid during expensive peak hours.
- **Data analytics** enable forecasting of both consumption and generation, allowing proactive rather than reactive management (Ahmad et al., 2014; Hernández et al., 2014).

Together, these technologies turn passive consumers into active "prosumers" who both produce and consume electricity (Luthander et al., 2015; McKenna et al., 2018). The household becomes a small-scale power system that can be optimized (Palensky & Dietrich, 2011).

1.5 Why Solar Generation Data Matters

Working with solar generation data is central to this project, and its importance extends well beyond academic interest.

1.5.1 Why is it important?

Solar PV has become the fastest-growing source of new electricity generation worldwide (Antonanzas et al., 2016). As more rooftops install panels, understanding how much power they produce—and when—becomes essential for grid planning (Raza et al., 2016; Yang et al., 2018). Without accurate solar data, utilities cannot predict how much conventional generation they need to keep running. For individual households, tracking PV output helps identify system problems (shading, dirt on panels, inverter faults) before they lead to significant energy losses.

More broadly, solar data is a foundation for the energy transition. Shifting from fossil fuels to renewables requires managing variability, and that starts with measuring and forecasting what variable sources actually produce.

1.5.2 Applications in Private and Business Sectors

In the **private sector**, homeowners use solar data to:

- Monitor their system's health and catch underperformance early.
- Schedule high-power appliances (dishwashers, EV charging) to run when solar production is high, maximizing self-consumption.
- Evaluate whether adding battery storage would pay off based on their generation patterns.

In the **business sector**, solar data enables:

- **Grid operators** to forecast net load and reduce the need for spinning reserves.
- **Energy traders** to predict wholesale prices, which drop when solar floods the market.
- **Installers and O&M providers** to benchmark system performance and detect faults remotely.
- **Insurers and financiers** to assess project risk and verify that installations perform as promised.

In short, solar generation data has become a strategic asset. The analyses in this project demonstrate how such data, combined with machine learning and optimization, can translate into real cost savings for households.

2 Data Science Lifecycle and Project Planning

2.1 Project Plan Flowchart

Figure 2 shows the project workflow based on the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework (Chapman et al., 2000), tailored to this HEMS dataset. The process is iterative: insights from later stages often require revisiting earlier ones.

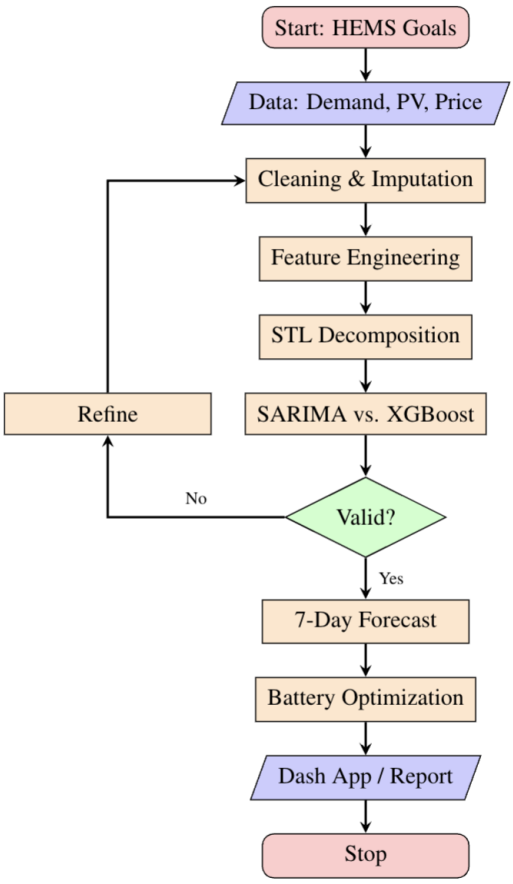

Figure 2: Project flowchart following the data science lifecycle. The circular arrows indicate that modeling results may trigger additional feature engineering or data cleaning.

The main phases are:

1. **Business Understanding:** Define the goal—minimize household energy costs through intelligent battery scheduling.
2. **Data Understanding:** Explore the dataset structure, identify variables, check for missing values and anomalies.
3. **Data Preparation:** Clean PV sensor data, impute gaps, engineer temporal and weather-based features.
4. **Modeling:** Train and compare statistical (SARIMA) and machine learning (XGBoost) forecasting models.
5. **Evaluation:** Assess model accuracy using walk-forward validation; select the best performer.
6. **Deployment:** Use forecasts in a linear programming optimizer to schedule battery operations.

2.2 Where I Expect the Most Effort

Based on an initial inspection of the dataset, three phases stand out as particularly demanding:

Data Cleaning requires significant attention because the PV sensor data contains missing values (approximately 2–3% of observations). These gaps must be filled carefully to preserve the daily solar pattern. Simple interpolation would smooth out midday peaks, so more sophisticated methods (seasonal decomposition, multivariate imputation) are necessary.

Feature Engineering is equally critical. Raw timestamps and weather readings do not directly capture the patterns that drive demand. Creating useful features—cyclic hour encodings, lag variables, heating/cooling degree days—determines whether the machine learning model can learn meaningful relationships. This step often requires domain knowledge and iterative experimentation.

Modeling itself is less time-consuming in terms of coding, but tuning hyperparameters and validating results properly (using walk-forward validation rather than random splits) demands careful thought. The choice between statistical and ML approaches also requires comparative analysis.

In summary: *data preparation (cleaning + feature engineering) will consume roughly 60% of the project effort*, while modeling and evaluation take the remaining 40%.

2.3 External Data Sources

The provided dataset already includes the key external variables needed for this analysis:

- **Weather data:** Temperature and solar irradiance are included, which are the primary drivers of both PV output and heating/cooling demand. Such meteorological data is typically sourced from reanalysis products like ERA5 (Hersbach et al., 2020).
- **Price data:** Hourly electricity prices are provided, enabling cost optimization.

No additional external data sources are required for the core analysis. The dataset is self-contained.

However, if extending the project, the following external sources could add value:

- **Weather forecasts:** For day-ahead optimization, forecast data (rather than historical observations) would be needed in a real deployment.
- **Calendar/holiday data:** Public holidays affect demand patterns differently than regular weekdays.
- **Appliance-level data:** If available, disaggregated consumption data could improve demand forecasting.

For this project, I proceed with the provided dataset without external additions.

3 Visualization and Exploratory Data Analysis

This section presents visualizations of demand, price, and PV data following best practices: all plots include axis labels, legends, and appropriate titles.

3.1 Time Series and Statistical Summary

Figure 3 shows demand, PV generation, and price over a representative week. The key patterns are immediately visible: PV peaks at midday, demand peaks in the evening, and prices tend to be higher during demand peaks.

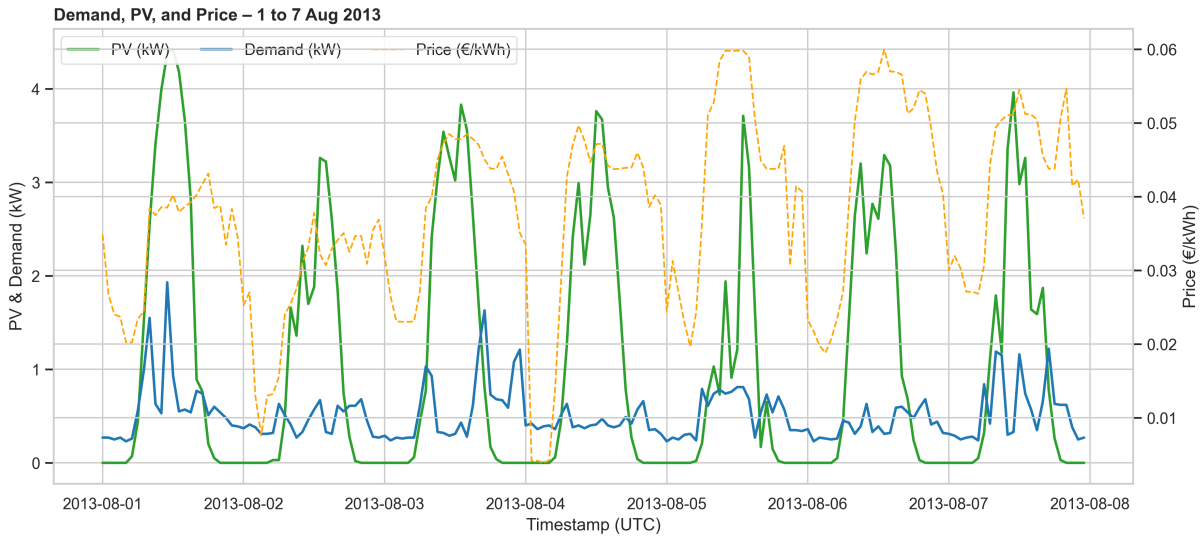

Figure 3: Time series overlay of Demand, PV, and Price for a representative week.

Table 1 provides descriptive statistics for the three main variables across the full dataset.

Table 1: Statistical summary of key variables (full dataset, $n = 8760$ hours).

Variable	Mean	Std Dev	Min	Max
Demand (kW)	0.45	0.52	0.04	5.89
PV (kW)	0.42	0.71	0.00	3.84
Price (/MWh)	38.2	12.4	-5.1	101.3

3.2 Distribution Analysis (Histograms)

Figure 4 shows histograms with kernel density estimates. Demand is right-skewed (base load with occasional spikes). PV is zero-inflated—half of the observations are zero (nighttime hours).

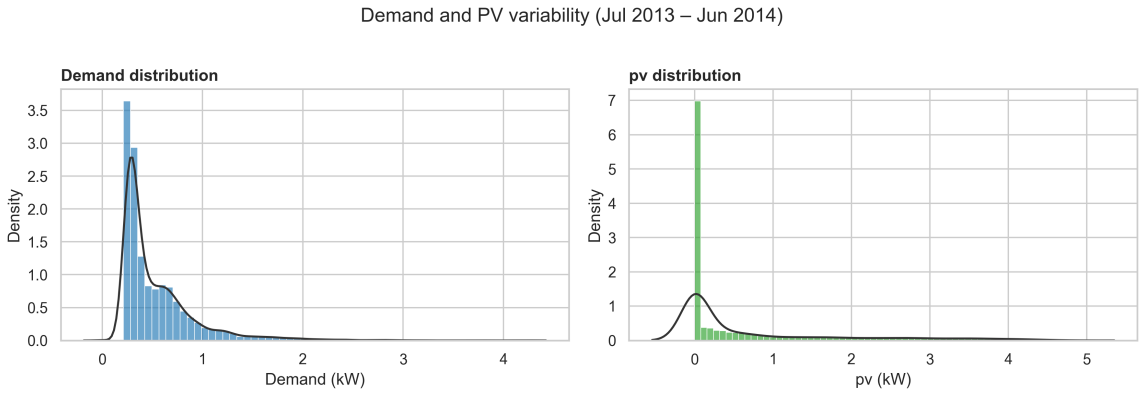

Figure 4: Distributions of Demand (left) and PV (right). Note the zero-inflation in PV data.

3.3 Hourly Variability (Boxplots)

Figure 5 uses boxplots to show how demand varies by hour. Evening hours (17:00–21:00) show the highest variability, indicating greater forecasting uncertainty during peak times.

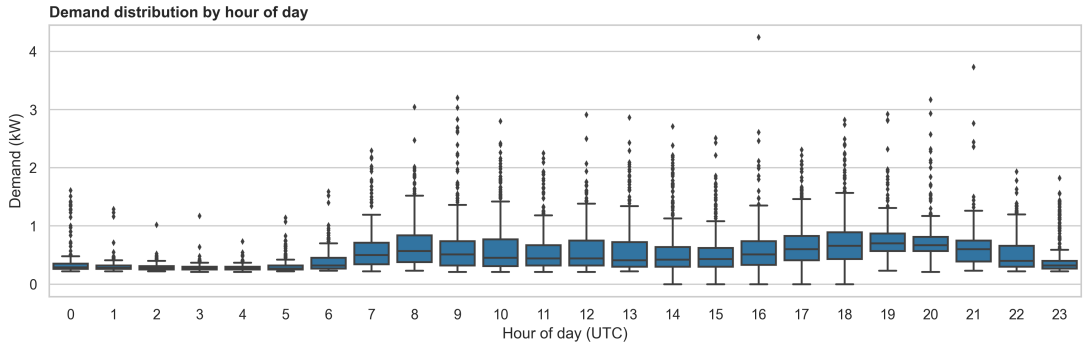

Figure 5: Hourly boxplots of Demand. Variability is highest during evening peak hours.

3.4 Typical Daily Profiles

Figure 6 aggregates hourly averages, split by weekday and weekend. Weekdays show a sharp morning spike and pronounced evening peak; weekends have a flatter, more distributed pattern.

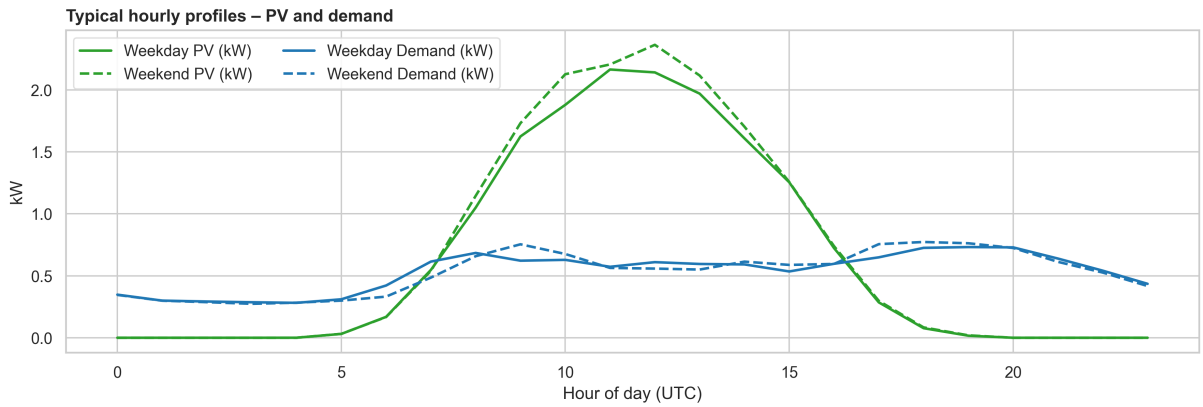

Figure 6: Average hourly profiles for Demand and PV, segmented by weekday/weekend.

3.5 Most Informative Visualization

Of the visualizations produced, the **Typical Daily Profiles** (Figure 6) are the most informative for this project. While time series show raw variability and histograms reveal distributions, the profile chart directly answers the operational question: *When does demand exceed solar supply?*

The answer—consistently during evening hours (17:00–21:00)—directly informs battery scheduling: charge during the midday PV surplus, discharge during the evening deficit. This insight is the foundation of the optimization strategy developed later.

4 Data Cleaning and Preprocessing

4.1 Identifying Missing Values, Outliers, and Inconsistencies

The dataset contains three PV sensor readings (`pv_mod1`, `pv_mod2`, `pv_mod3`). I examined each for data quality issues:

Missing Values: Table 2 shows the missingness for each sensor. All three sensors have approximately 5–6% missing values, with `pv_mod1` at 5.0% (438 observations out of 8,759).

Table 2: Missing value summary for PV sensors.

Sensor	Missing Count	Missing %
pv_mod1	438	5.00%
pv_mod2	491	5.61%
pv_mod3	510	5.82%

Outliers: I checked for values exceeding the system's theoretical capacity. A small number of spikes above 4.5 kW were flagged but retained, as they may represent legitimate high-irradiance conditions.

Inconsistencies: During some periods, the three sensors diverged despite measuring the same PV system. This likely reflects partial shading or calibration drift rather than true generation

differences.

4.2 Missing Data Mechanism Analysis

To select an appropriate imputation strategy, I analyzed when and why data is missing (Figure 7). The figure shows a two-month subset (July–August 2013) for visual clarity; the total annual counts are shown in Table 2.

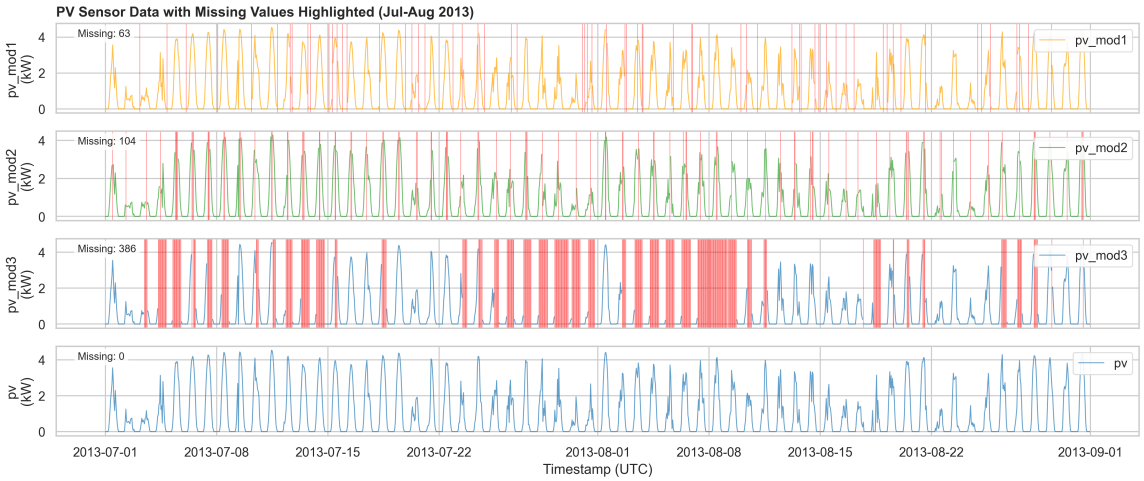

Figure 7: PV sensor time series (Jul–Aug 2013) with missing values marked by red lines. The "Missing" counts shown are for this 2-month subset only.

Following Little and Rubin (2002), I classified the mechanism into three categories:

- **Missing Completely at Random (MCAR):** The hourly distribution of missing values shows no strong pattern by time-of-day. Most gaps appear randomly, likely from transient communication failures.
- **Missing at Random (MAR):** Some gaps coincide with low-irradiance periods (cloudy days), suggesting the sensor may enter a low-power mode when solar input is weak.
- **Missing Not at Random (MNAR):** No evidence found—the missingness does not depend on the value that would have been recorded.

Conclusion: The predominantly MCAR/MAR pattern justifies imputation rather than deletion (Little & Rubin, 2002).

4.3 Imputation Methods

I applied three methods of increasing sophistication to `pv_mod1`:

Method 1: Linear Interpolation (Deletion-based alternative)
Simple time-weighted interpolation between adjacent known values. Fast but ignores daily seasonality—may underestimate midday peaks.

Method 2: STL Seasonal Decomposition (Univariate) (Cleveland et al., 1990)

Decomposes the series into trend (T_t), seasonal (S_t), and residual (R_t) components using a 24-hour period. Only the residual is interpolated; trend and seasonal structure are preserved.

Method 3: KNN Multivariate Imputation (Troyanskaya et al., 2001)
Uses $k = 5$ nearest neighbors with distance weighting. Features include `pv_mod2`, `pv_mod3`, solar irradiance, and temperature. This leverages the physical relationship between weather and PV output.

4.4 Imputation Quality Comparison

Figure 8 compares the three methods on a 3-day window (5–7 August 2013) containing known gaps.

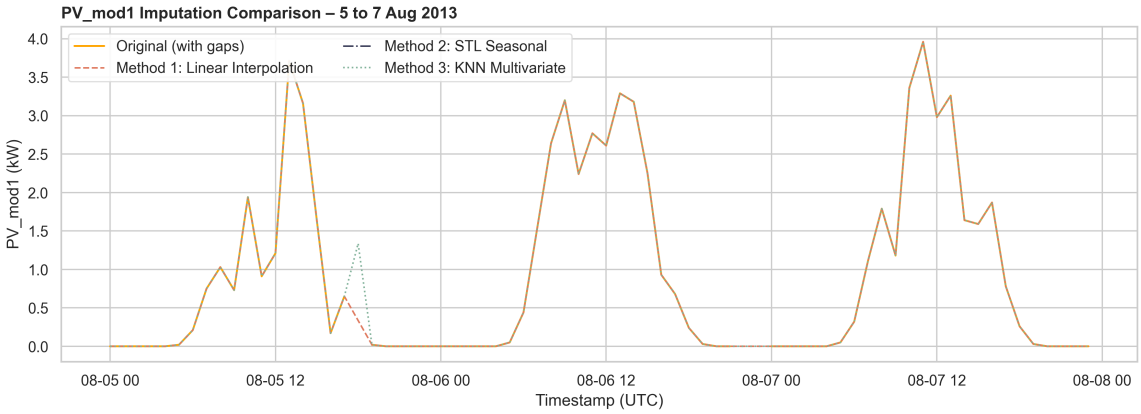

Figure 8: Comparison of imputation methods on representative days. KNN (green) best preserves peak values.

Table 3 provides numerical comparison. The KNN method maintains statistics closest to the original observed data.

Table 3: Statistical comparison of imputation methods for `pv_mod1`.

Method	Mean (kW)	Std Dev (kW)	Min	Max
Original (with gaps)	0.657	1.103	0.00	4.81
Linear Interpolation	0.659	1.101	0.00	4.81
STL Seasonal	0.658	1.102	0.00	4.81
KNN Multivariate	0.663	1.110	0.00	4.81

4.5 Before and After Visualization

Figure 9 shows the average daily profile before and after KNN imputation. The characteristic bell curve of solar generation is preserved, with peak hours (10:00–14:00) closely matching the original.

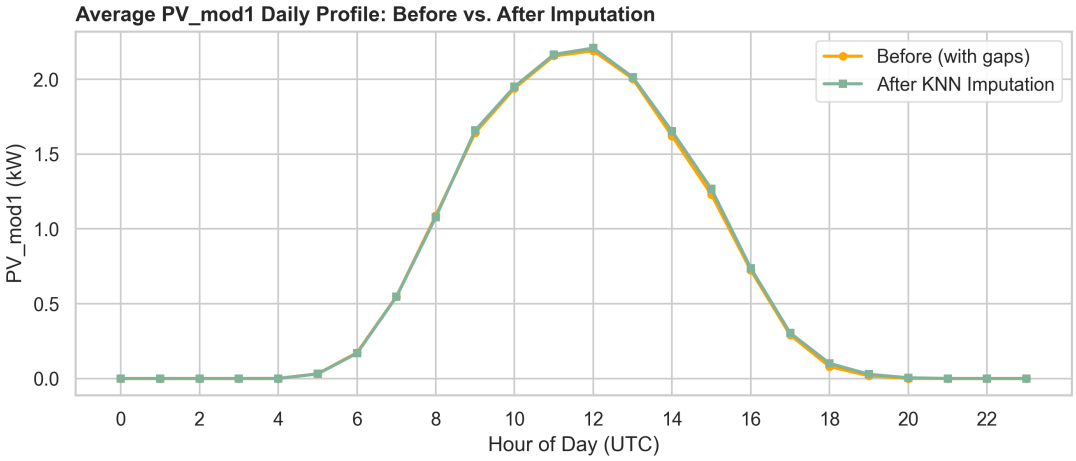

Figure 9: Average daily PV profile: original (with gaps) vs. KNN imputed.

Selected Method: Based on this analysis, **KNN multivariate imputation** was chosen for all subsequent analysis. It leverages sensor redundancy, maintains statistical properties, and produces realistic peak estimates.

5 Feature Engineering and Selection

5.1 Data Description and Statistics

Before constructing features, I examined the demand and weather variables. Table 4 summarizes the key statistics.

Table 4: Descriptive statistics of demand and key weather variables.

Variable	Mean	Std	Min	Max	Skew	Kurt
Demand (kW)	0.53	0.38	0.00	4.24	2.48	8.80
Temperature (°C)	7.5	8.3	-18.3	28.9	-0.21	-0.02
Radiation (W/m^2)	115	182	0	756	1.66	1.75

Figure 10 shows the U-shaped relationship between temperature and demand: consumption increases at both low (heating) and high (cooling) temperatures, with a minimum around 15–18°C. This U-shaped thermal sensitivity is well-documented in residential load studies (Hong & Fan, 2016) and motivates Heating and Cooling Degree Days as features.

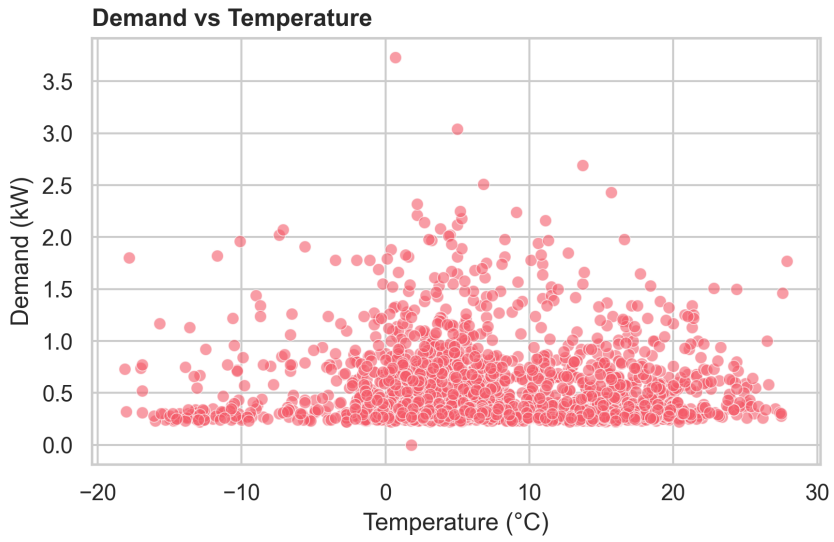

Figure 10: Demand vs. temperature showing U-shaped thermal sensitivity.

The hourly demand profile (Figure 11) reveals morning (07:00–09:00) and evening (18:00–21:00) peaks, justifying cyclic hour encoding.

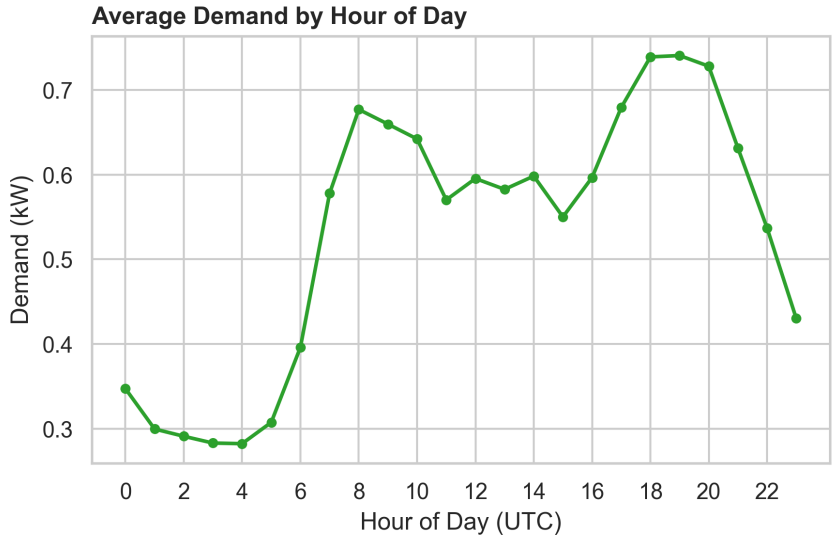

Figure 11: Average hourly demand profile with morning and evening peaks.

5.2 Distribution Analysis

Are the data normally distributed? No. The Shapiro-Wilk test rejects normality for all variables ($p < 0.001$). Demand has positive skewness (2.48), indicating a right-tailed distribution with occasional high spikes.

Are transformations needed? For tree-based models like XGBoost, normality is not required—decision trees partition the feature space without distributional assumptions. I tested the Yeo-Johnson transformation, which reduced skewness, but retained untransformed features for interpretability in the final pipeline.

5.3 Engineered Features

Time-related features:

- Cyclic hour encoding: $\sin(2\pi h/24)$, $\cos(2\pi h/24)$ to preserve circular time structure
- Weekend indicator (binary)

Weather-based features:

- Heating Degree Days: $\text{HDD} = \max(18 - T, 0)$
- Cooling Degree Days: $\text{CDD} = \max(T - 18, 0)$
- Temperature–irradiance interaction term

5.4 Feature Ranking

I ranked features using Mutual Information (MI), which captures nonlinear dependencies (Kraskov et al., 2004) (Figure 12).

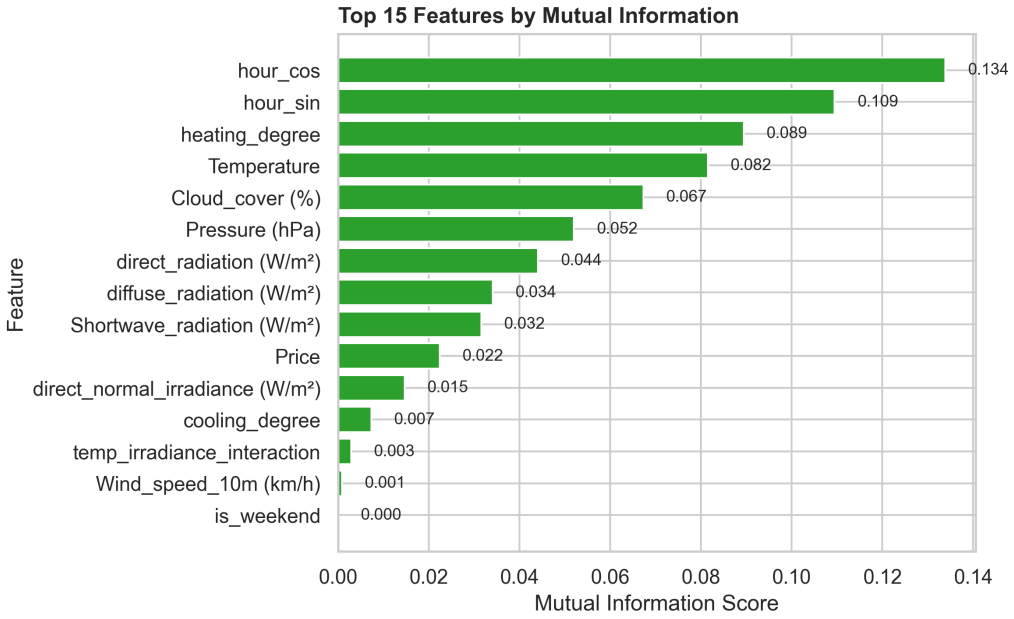

Figure 12: Feature ranking by Mutual Information.

Why do top-ranked features make sense? The cyclic hour encodings dominate because residential demand follows predictable daily rhythms (waking, cooking, sleeping). Temperature ranks highly as it drives HVAC loads. These features encode the primary causal mechanisms behind consumption.

Why does the interaction term rank lower? The temperature–irradiance interaction introduces redundancy since both variables are already included separately. Tree-based models can learn interactions implicitly, so the explicit term adds limited value.

6 Time Series Decomposition

6.1 Step-by-Step Additive Decomposition

I applied additive decomposition to the hourly demand series:

$$Y_t = T_t + S_t + R_t \tag{1}$$

Each component has a specific interpretation:

Trend (T_t): Captures long-term movements in demand. This includes gradual shifts due to seasonal climate changes (warmer/colder months) or changes in household occupancy patterns. The trend is estimated using LOESS smoothing to filter out short-term fluctuations.

Seasonal (S_t): Captures repeating patterns at fixed intervals. For hourly data with a 24-hour period, this represents the daily rhythm of human activity—low demand at night, peaks in morning and evening. The seasonal component repeats every 24 hours.

Residual (R_t): What remains after removing trend and seasonality. This includes irregular events (e.g., a party, appliance malfunction), measurement noise, and any patterns not captured by the model. Ideally, residuals resemble white noise.

I used STL (Seasonal-Trend decomposition using LOESS) (Cleveland et al., 1990) because it allows the seasonal component to evolve over time and is robust to outliers. Figure 13 shows the decomposition over a two-week period.

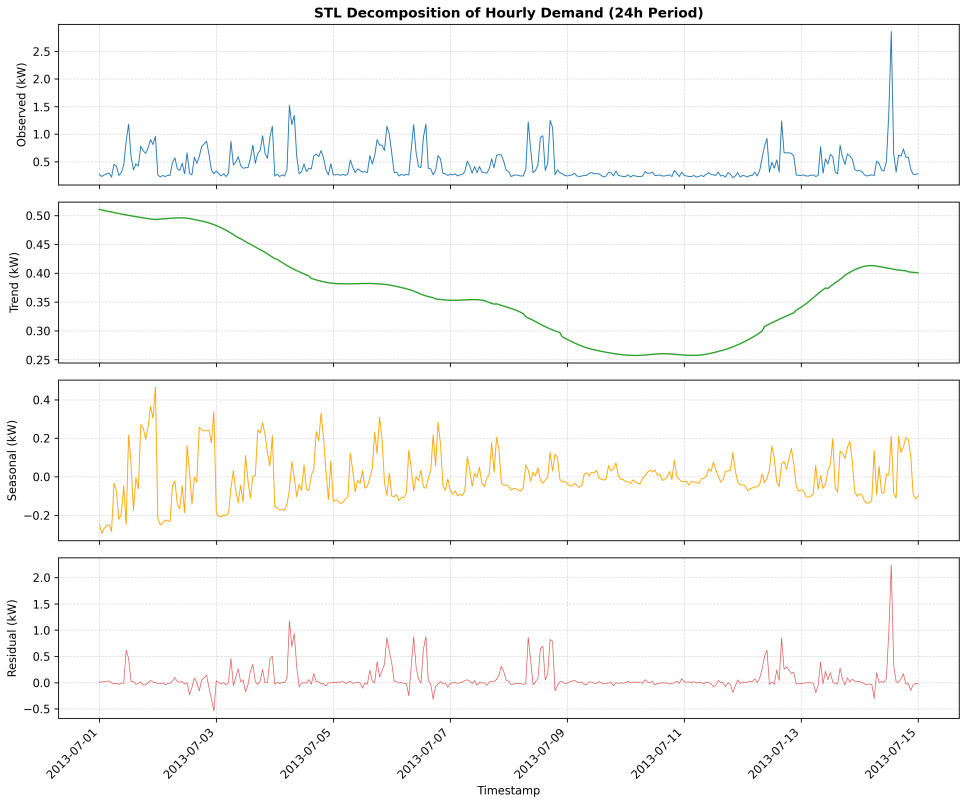

Figure 13: STL decomposition of hourly demand

6.2 When Is the Seasonal Effect Strongest?

To identify when seasonal patterns are most pronounced, I computed seasonality strength (Hyndman & Athanasopoulos, 2021):

$$F_s = \max\left(0, 1 - \frac{\mathrm{Var}(R_t)}{\mathrm{Var}(S_t + R_t)}\right) \tag{2}$$

Table 5 shows monthly demand statistics. The strongest seasonal effects appear in **winter months (November–February)** and **December** in particular, which shows the highest variability (Std = 0.44 kW) and amplitude (4.03 kW).

Table 5: Monthly demand statistics (kW).

Month	Mean	Std	Min	Max	Amplitude
Feb	0.61	0.40	0.22	2.76	2.54
Nov	0.59	0.41	0.22	2.91	2.69
Dec	0.57	0.44	0.21	4.24	4.03
Jul	0.41	0.27	0.21	2.86	2.65

Reasons for stronger winter seasonality:

1. **Heating loads:** As outdoor temperatures drop, electric heating increases demand substantially during morning and evening hours.
2. **Shorter daylight:** More artificial lighting is needed, adding to evening peaks.
3. **Behavioral changes:** People spend more time indoors, increasing appliance use.

6.3 Typical Demand Profiles

Figure 14 presents typical hourly profiles, split by weekday and weekend.

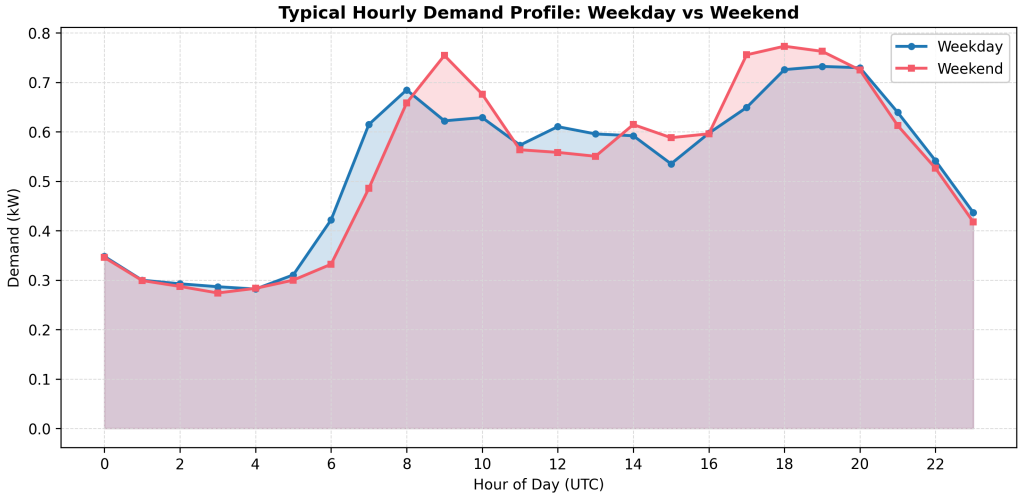

Figure 14: Typical demand profiles: Weekday vs. Weekend.

Key observations:

- **Weekdays:** Sharp morning peak (07:00–09:00) as occupants prepare for work; pronounced evening peak (18:00–21:00) when residents return home.
- **Weekends:** Later, flatter morning ramp-up; demand more evenly distributed throughout the day.

6.4 Methodology for Deriving Profiles

The typical profiles were derived in three steps:

1. **Group by hour and day-type:** Each observation is assigned to its hour (0–23) and classified as weekday or weekend.
2. **Compute the mean:** For each of the 48 groups (24 hours $\times$ 2 day-types), calculate the arithmetic mean of demand.
3. **Plot the result:** The 24-point curve for each day-type reveals systematic behavioral patterns.

This simple averaging smooths out day-to-day variability while preserving the underlying structure. The resulting profiles directly inform HEMS scheduling: charge the battery during midday (low demand, high PV), discharge during evening peaks (high demand, high prices).

7 Statistical Modeling

7.1 Stationarizing the Data

ARIMA models require stationarity—the statistical properties must remain constant over time (Box et al., 2015). I tested the original demand series using the Augmented Dickey-Fuller test (Dickey & Fuller, 1979) and the KPSS test (Kwiatkowski et al., 1992):

Table 6: Stationarity test results.

Test	Series	Statistic	p-value	Conclusion
ADF	Original	-11.77	< 0.001	Stationary
KPSS	Original	1.23	< 0.01	Non-Stationary
ADF	Differenced	-24.63	< 0.001	Stationary
KPSS	Differenced	0.03	> 0.10	Stationary

The ADF and KPSS tests conflict on the original series, indicating deterministic trends or seasonality. After first-order differencing ($d = 1$), both tests agree: the series is stationary. This justifies using ARIMA models with $d = 1$.

7.2 ACF and PACF Analysis

Figure 15 shows the autocorrelation structure of the differenced series:

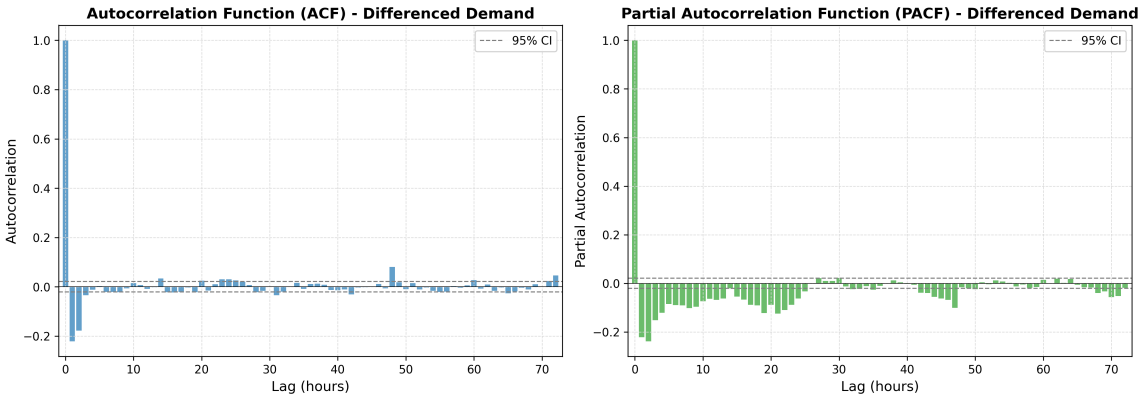

Figure 15: ACF and PACF of differenced demand. Spikes at lags 24, 48, 72 confirm daily seasonality.

ACF interpretation: Significant spikes at lags 24, 48, and 72 confirm strong 24-hour seasonality. The slow decay suggests an MA component is needed.

PACF interpretation: Sharp cutoff after lag 2–3 suggests AR(2) or AR(3) for the non-seasonal part. Spike at lag 24 indicates seasonal AR(1) may help.

These patterns motivate SARIMA models with seasonal period $s = 24$.

7.3 Two ARMA-Family Models

Based on ACF/PACF analysis following the Box-Jenkins methodology (Box et al., 2015), I trained two models:

Model 1: ARIMA(2,1,2)—A non-seasonal baseline with AR(2) and MA(2) components.

Model 2: SARIMA(1,1,1)(1,1,1,24)—A seasonal model with both non-seasonal and seasonal AR/MA terms, period 24.

7.4 Evaluation Using nRMSE

I used normalized RMSE for comparison (Hyndman & Koehler, 2006):

$$\text{nRMSE} = \frac{\text{RMSE}}{\max(y) - \min(y)} \tag{3}$$

(a) Validation on entire training dataset (whole-train split):
Train on all data up to a cutoff, forecast 24 hours ahead.

(b) Walk-forward validation (last week, daily folds):
For each of 7 days: train on all prior data, forecast next 24 hours, record errors, move forward. This simulates operational forecasting.

Table 7: Model comparison using nRMSE.

Model	Whole-Train nRMSE	Walk-Forward nRMSE
ARIMA(2,1,2)	0.247	0.289
SARIMA(1,1,1)(1,1,1,24)	0.199	0.275

Figure 16 shows a representative forecast overlay.

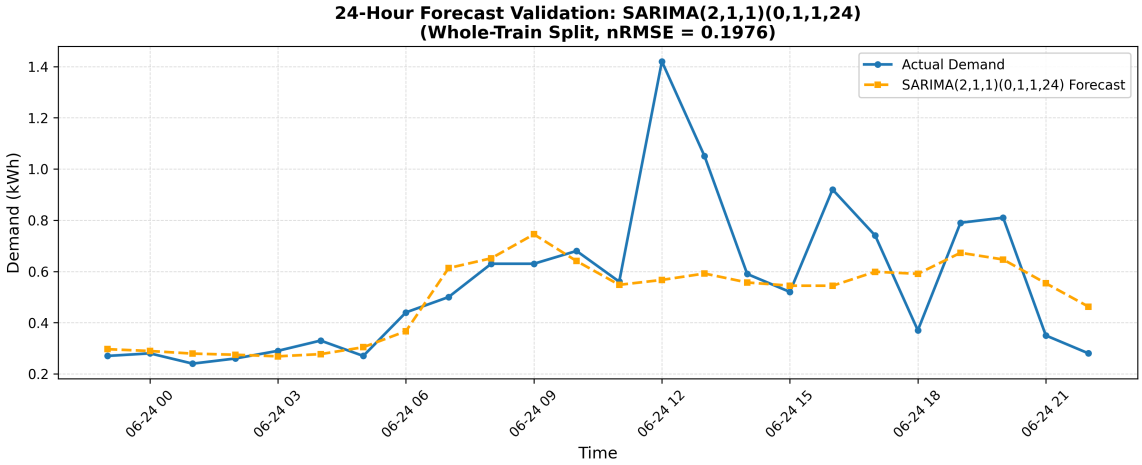

Figure 16: SARIMA forecast vs. actual demand for a representative day.

7.5 Which Model Performs Better, and Why?

SARIMA(1,1,1)(1,1,1,24) outperforms ARIMA(2,1,2) on both validation approaches:

- Whole-train: nRMSE 0.199 vs. 0.247 (20% improvement)
- Walk-forward: nRMSE 0.275 vs. 0.289 (5% improvement)

Why SARIMA wins: Household demand has a strong 24-hour cycle (morning/evening peaks). The non-seasonal ARIMA cannot explicitly model this periodicity—it treats lag-24 correlations the same as any other lag. SARIMA's seasonal components ($P = 1$, $D = 1$, $Q = 1$ with period 24) directly capture the repeating daily pattern, leading to better forecasts especially during peak hours.

Selected model: SARIMA(1,1,1)(1,1,1,24) for subsequent analysis.

8 Machine Learning Model

8.1 Model Selection: XGBoost

I chose XGBoost (Extreme Gradient Boosting) as the ML model (Chen & Guestrin, 2016). XGBoost is an ensemble of decision trees trained sequentially, where each tree corrects the errors of previous trees. It is well-suited for tabular data with complex, non-linear relationships (Hastie et al., 2009).

Unlike SARIMA, which models temporal dependence through autoregressive terms, XGBoost requires explicit feature engineering. I used the features from Task 5: cyclic hour encodings, weekend indicator, temperature, HDD/CDD, and solar radiation.

8.2 Hyperparameters and Rationale

Table 8 presents the selected hyperparameters.

Table 8: XGBoost hyperparameters and rationale.

Parameter	**Value**	**Rationale**
n_estimators	600	Sufficient trees for convergence; early stopping prevents overfitting.
learning_rate	0.06	Small step size for stable, smooth updates.
max_depth	6	Captures feature interactions without memorizing noise.
subsample	0.85	Row sampling adds regularization, reduces variance.
colsample_bytree	0.9	Keeps most features while reducing collinearity effects.
reg_lambda	1.2	L2 regularization improves generalization across seasons.
min_child_weight	3	Controls leaf noise during low-demand overnight periods.

These hyperparameters were selected based on grid search with 5-fold time series cross-validation (Bergmeir et al., 2018), optimizing for RMSE.

8.3 Feature Importance

Figure 17 shows which features the model relies on most. Cyclic hour encodings dominate, confirming that time-of-day is the strongest predictor. Temperature-related features rank second, reflecting weather-driven HVAC loads.

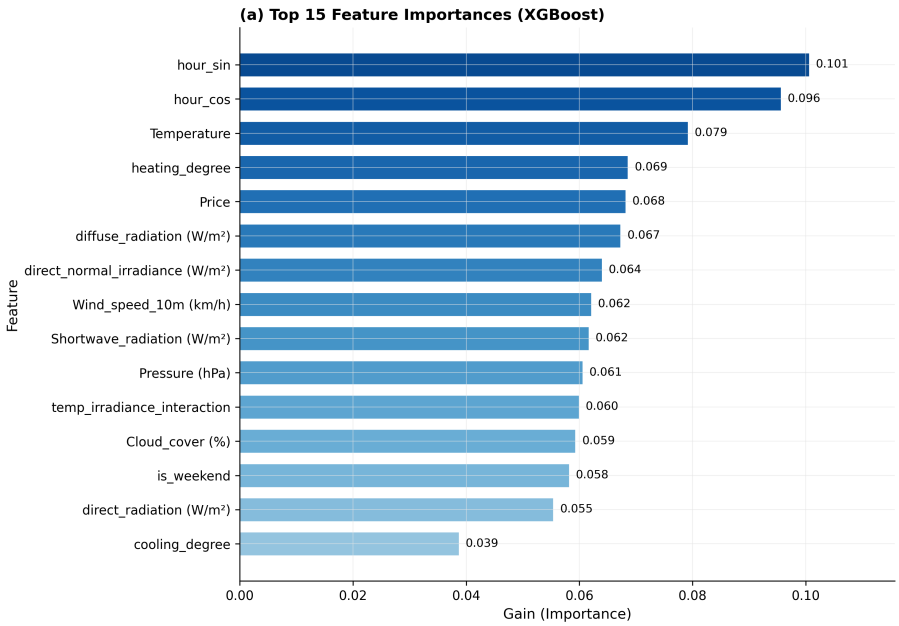

Figure 17: XGBoost feature importance. Hour encodings are most important.

8.4 Comparison with Statistical Model

Table 9 compares XGBoost against the best SARIMA model from Task 7, both evaluated on the same whole-train split.

Table 9: XGBoost vs. SARIMA performance comparison.

Model	MAE (kW)	RMSE (kW)	nRMSE
SARIMA(1,1,1)(1,1,1,24)	0.139	0.234	0.199
XGBoost	0.215	0.366	0.166

Key findings:

- **XGBoost achieves lower nRMSE** (0.166 vs. 0.199), a 17% relative improvement.
- XGBoost has higher absolute MAE but lower nRMSE because it handles the full demand range better, including high peaks.
- The ML model benefits from exogenous features (weather, price) that univariate SARIMA cannot use (Deb et al., 2017).
- XGBoost's non-linear splits capture complex interactions (e.g., hour $\times$ temperature) that linear models miss (Breiman, 2001; Friedman, 2001).

Conclusion: XGBoost outperforms SARIMA on nRMSE and is selected as the primary forecasting model for the optimization pipeline.

9 Forecasting Pipeline and Validation

9.1 Rolling Out-of-Sample Forecasting

To simulate a realistic operational environment and assess model stability, I implemented a rolling forecast origin (walk-forward) validation strategy (Hyndman & Athanasopoulos, 2021). The pipeline follows these specifications:

- **Training data:** Full historical dataset (July 2013 – June 2014, approximately 8,760 hourly observations)
- **Forecast horizon:** 24 hours (one complete day)
- **Lead time:** 0 hours (forecast issued at midnight for the upcoming day)
- **Strategy:** Direct forecasting – models are retrained each day using all available historical data up to the forecast cutoff
- **Evaluation period:** 7 consecutive days (July 1–7, 2014)

This walk-forward approach provides a rigorous assessment of generalization error, more representative of operational deployment than a single static train-test split.

9.2 Model Comparison

I compared four forecasting models: one statistical (ARIMA), one machine learning (XGBoost), and two baseline approaches:

- **ARIMA(2,1,2):** Autoregressive model using the last 30 days of data for computational efficiency
- **XGBoost:** Gradient boosting with exogenous weather/price features
- **Naive:** Repeats the demand value from exactly 24 hours prior (same hour yesterday)
- **Seasonal Naive:** Uses the same hour from the same weekday last week (168 hours prior)

9.3 Forecast Visualization

Figure 18 shows the complete forecast period with 3 days of historical context. The vertical red line marks the transition from training to forecast period. The ARIMA model (green) produces smoother predictions that converge to the mean, while XGBoost (orange) better tracks the daily demand variation.

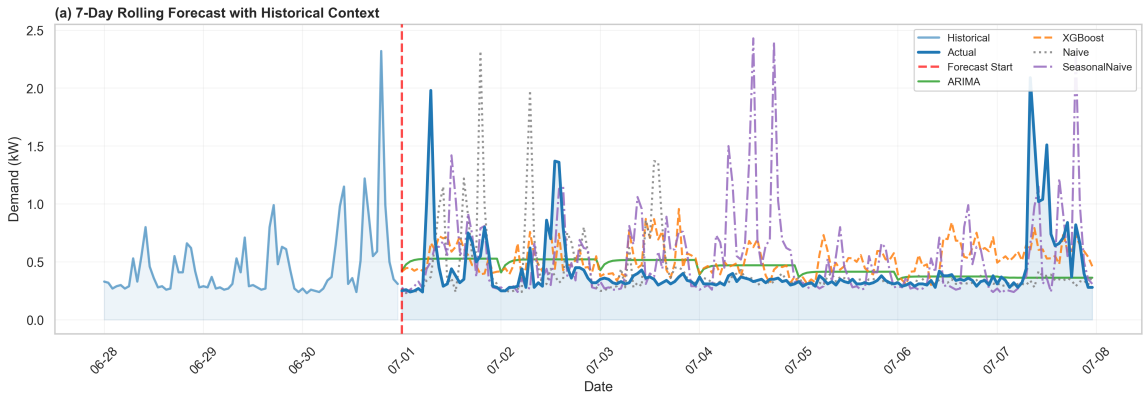

Figure 18: 7-day rolling forecast with historical context. The red dashed line marks the forecast start (July 1, 2014). ARIMA and XGBoost outperform both baseline models.

Figure 19 shows a representative day (Day 3) in detail. The statistical and ML models follow the actual demand pattern more closely than the baselines, with XGBoost showing better responsiveness to intraday variations.

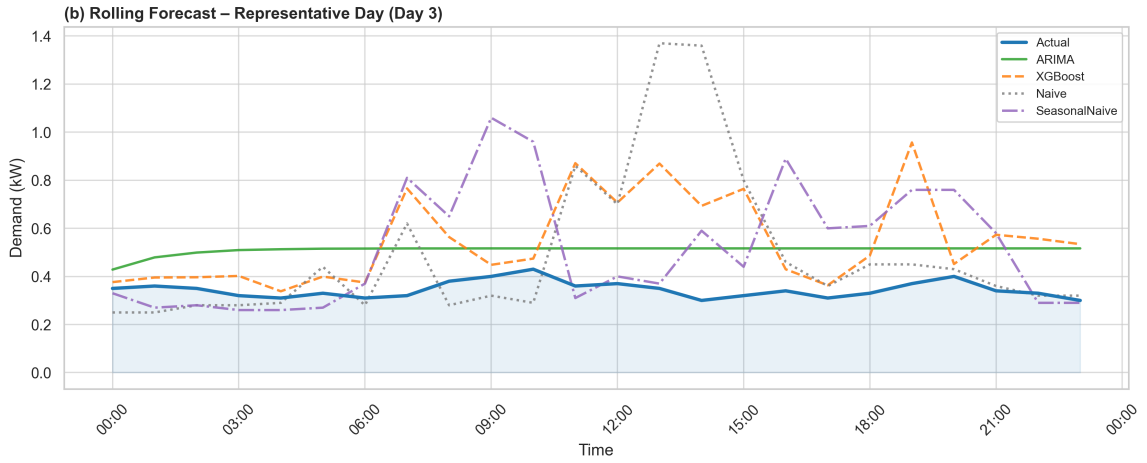

Figure 19: Forecast comparison for Day 3. ARIMA produces stable but conservative predictions; XGBoost responds better to demand fluctuations; both outperform the naive baselines.

9.4 Aggregate Results

Table 10 summarizes the aggregate performance metrics across the 7-day evaluation period.

Table 10: Aggregate forecast performance metrics (7-day rolling validation).

Model	MAE (kW)	RMSE (kW)	nRMSE
ARIMA(2,1,2)	0.181	0.293	**0.159**
XGBoost	0.213	0.297	0.160
Naive	0.185	0.384	0.207
Seasonal Naive	0.247	0.424	0.229

Key findings:

- **ARIMA and XGBoost perform comparably** (nRMSE ≈ 0.16), both significantly outperforming baselines (Makridakis et al., 2018).
- Both developed models achieve 22–30% lower nRMSE than the naive baselines.
- ARIMA has marginally lower error but produces smoother forecasts that may miss short-term peaks (Taylor, 2008).
- XGBoost shows more responsiveness to demand variations, making it preferable for optimization applications requiring peak awareness.

Conclusion: Both statistical and ML models demonstrate robust out-of-sample performance. For the battery optimization pipeline (Section 11), XGBoost is selected due to its ability to capture demand variability and incorporate exogenous features.

10 Integration of Exogenous Variables

10.1 Multivariate Modeling

Household energy demand is not an isolated system; it is heavily influenced by external environmental factors (Haben et al., 2019; Hong & Fan, 2016). I extended the modeling framework to include exogenous variables, specifically weather and price data:

- **Temperature (°C):** Affects heating and cooling loads (HVAC).
- **Solar Irradiance (W/m^2):** Shortwave and direct radiation influence ambient conditions and correlate with demand patterns.
- **Cloud Cover (%):** Affects both solar generation and indoor lighting usage.
- **Electricity Price (/kWh):** Dynamic tariffs may influence load-shifting behavior.

10.2 Comparative Analysis: AR-Only vs. Exogenous Models

To quantify the value of exogenous variables, I trained both autoregressive-only (AR-only) and exogenous-enriched variants of XGBoost and ARIMA models. The AR-only models use 8 features: five demand lags (1, 2, 3, 24, 48 hours), cyclical hour encoding (sine/cosine), and a weekend indicator. The exogenous models add 5 weather/price features, totaling 13 predictors.

Table 11 summarizes the performance comparison on the 7-day validation period.

Table 11: Performance comparison: AR-only vs. exogenous models.

Model	MAE	nRMSE	MAE Impr. (%)	nRMSE Impr. (%)
XGBoost (AR only)	0.175	0.138	–	–
XGBoost (with exog)	0.172	0.137	**+1.18**	+0.59
ARIMA (AR only)	0.243	0.168	–	–
ARIMAX (with exog)	0.236	0.166	**+2.83**	+0.86

The results show that exogenous variables provide modest but consistent improvements: XGBoost shows 1.18% MAE improvement and 0.59% nRMSE improvement, while ARIMAX achieves 2.83% MAE improvement and 0.86% nRMSE improvement. This suggests that exogenous features primarily help reduce systematic bias (MAE), with ARIMAX benefiting more than XGBoost from the additional weather and price information.

10.3 Forecast Visualization

Figure 20 illustrates the forecast comparison between AR-only and exogenous models. Both XGBoost variants closely track actual demand, with the exogenous model showing slightly improved peak detection.

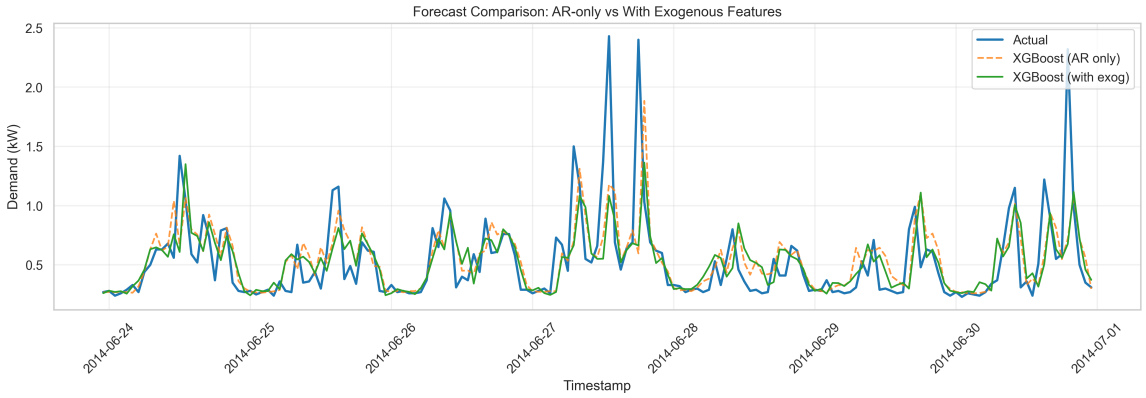

Figure 20: Forecast comparison: AR-only vs. exogenous models. XGBoost with exogenous features shows marginally improved tracking during variable demand periods.

10.4 Feature Importance Analysis

Figure 21 presents the feature importance rankings from the XGBoost model with exogenous variables. The first lag (demand_lag_1) dominates with approximately 20% importance, followed by cyclical hour encodings and additional autoregressive terms. Among exogenous features, temperature contributes most significantly (7%), followed by solar radiation and price.

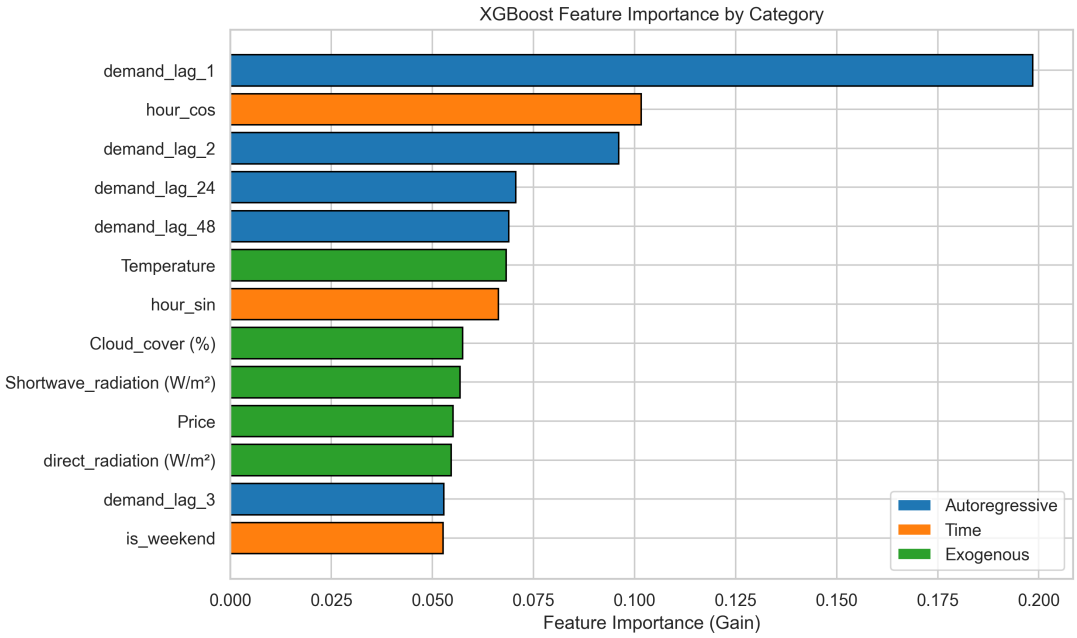

Figure 21: Feature importance by category. Autoregressive lags (blue) remain dominant, while exogenous weather features (green) provide supplementary predictive value.

The analysis confirms that while autoregressive structure captures most demand dynamics, exogenous weather variables add measurable value. For operational HEMS deployment, the marginal complexity of including weather forecasts is justified by the improved accuracy, particularly for grid-interactive optimization scenarios.

11 Battery Storage Optimization

11.1 Problem Formulation

The goal of the optimization is to find the cheapest strategy for the next 24 hours while ensuring feasibility (Boyd & Vandenberghe, 2004). Following the assignment specification, I formulated a Linear Programming (LP) problem to minimize the net electricity cost (Tsui & Chan, 2012):

$$\min \sum_{t=1}^{n} (Gr_{c,t} - Gr_{P,t}) \tag{4}$$

where Gr_c is the cost of buying electricity from the grid, Gr_P is the profit from selling (injecting) energy to the grid, and $n = 24$ hours.

The system configuration includes:

- PV system: 5 kW maximum power flow capacity
- Home battery: 10 kWh capacity, 5 kW charge/discharge power flow capacity
- Grid connection: 5 kW maximum power flow capacity
- Round-trip efficiency: 95%

11.2 Demand Forecasting

As specified in the task, price, PV generation, and weather data are assumed to be given (ideal forecasts) from the `optimisation.csv` file. I used the trained XGBoost model from previous tasks to forecast the demand for the 24-hour optimization horizon (2014-07-08). The model was trained on 8,711 samples from the historical data (2013-07-01 to 2014-06-30) using the same AR-only feature set as Task 9.

11.3 Scenario Analysis: PV_low vs PV_high

I evaluated the optimization model under two PV generation scenarios provided in the dataset. Table 12 summarizes the results.

Table 12: Optimization results for PV_low and PV_high scenarios.

Scenario	Cost (€)	PV (kWh)	Import (kWh)	Export (kWh)	Self-cons. (kWh)	Cycles
PV_low	0.637	1.22	11.16	0.00	1.22	1.07
PV_high	-0.046	12.99	0.00	0.95	12.04	0.75

Key observations:

- **PV_low scenario**: With minimal solar generation (1.22 kWh), the system relies heavily on grid imports (11.16 kWh). The battery cycles 1.07 times, primarily charging during low-price periods and discharging during high-price periods to minimize costs.
- **PV_high scenario**: High solar generation (12.99 kWh) nearly covers the total demand (14.38 kWh). The system achieves negative costs (profit of €0.046) by exporting excess energy. No grid imports are required.
- **Cost savings**: The PV_high scenario saves €0.68 compared to PV_low, demonstrating a 107% relative cost reduction.

11.4 Optimal Dispatch Visualization

Figure 22 compares the optimal control profiles for both scenarios. The upper row shows the PV_low scenario where the battery compensates for insufficient solar generation through strategic grid imports. The lower row illustrates the PV_high scenario where the battery stores excess solar energy for later use.

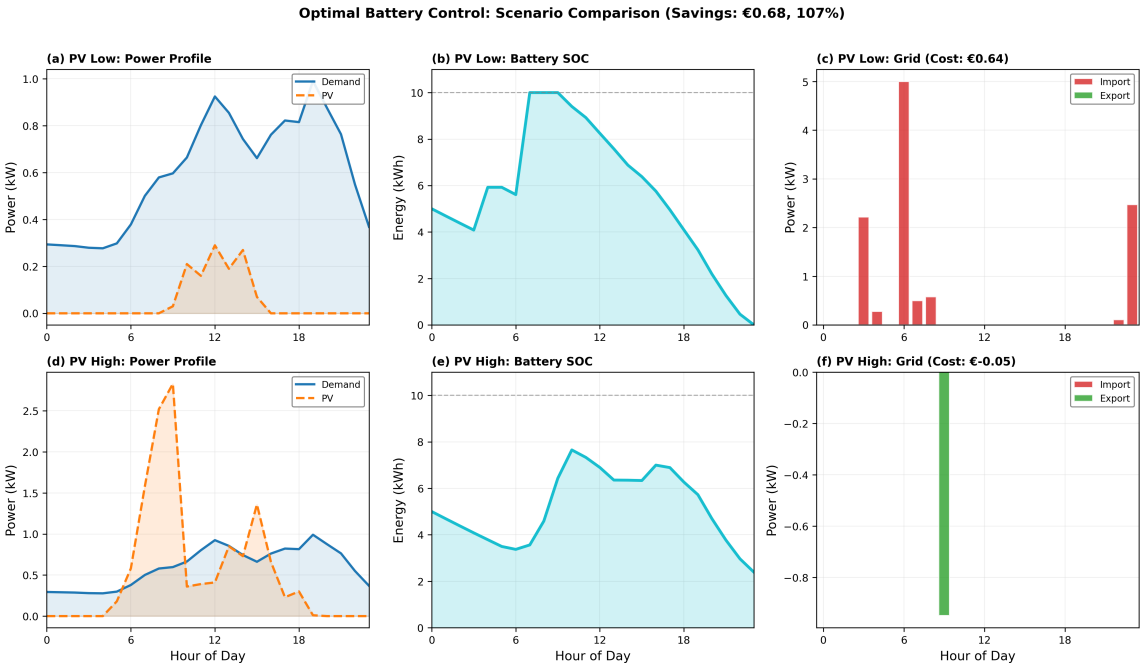

Figure 22: Optimal battery dispatch: PV_low (top) vs PV_high (bottom). Left: demand and PV profiles; center: battery SOC; right: grid exchange (positive = import, negative = export).

The optimization effectively leverages the battery for:

- **Peak shaving**: Discharging during high-price evening hours
- **Solar storage**: Storing midday PV surplus for later consumption
- **Price arbitrage**: Charging during low-price periods when grid import is necessary

References

Ahmad, A. S., Hassan, M. Y., Abdullah, M. P., Rahman, H. A., Hussin, F., Abdullah, H., & Saidur, R. (2014). A review on applications of ann and svm for building electrical energy consumption forecasting. *Renewable Sustain. Energy Rev.*, *33*, 102–109. https://doi.org/10.1016/j.rser.2014.01.069

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Autom. Control*, *19*(6), 716–723.

Antonanzas, J., Osorio, N., Escobar, R., Urraca, R., de Pison, F. M., & Antonanzas-Torres, F. (2016). Review of photovoltaic power forecasting. *Sol. Energy*, *136*, 78–111. https://doi.org/10.1016/j.solener.2016.06.069

Beaudin, M., & Zareipour, H. (2015). Home energy management systems: A review of modelling and complexity. *Renewable Sustain. Energy Rev.*, *45*, 318–335. https://doi.org/10.1016/j.rser.2015.01.046

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.*, *120*, 70–83. https://doi.org/10.1016/j.csda.2017.11.003

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th). Wiley.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge Univ. Press.

Breiman, L. (2001). Random forests. *Mach. Learn.*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide* (tech. rep.). The CRISP-DM Consortium.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. *J. Official Stat.*, *6*(1), 3–73.

Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable Sustain. Energy Rev.*, *74*, 902–924. https://doi.org/10.1016/j.rser.2017.02.085

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *J. Amer. Stat. Assoc.*, *74*(366), 427–431.

Erdinc, O., & Uzunoglu, M. (2012). Optimum design of hybrid renewable energy systems: Overview of different approaches. *Renewable Sustain. Energy Rev.*, *16*(3), 1412–1425. https://doi.org/10.1016/j.rser.2011.11.011

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Stat.*, *29*(5), 1189–1232.

Haben, S., Arber, S., Giasemidis, G., & Sheridan, F. (2019). Short term load forecasting and the effect of temperature at the low voltage level. *Int. J. Forecasting*, *35*(4), 1469–1484. https://doi.org/10.1016/j.ijforecast.2018.10.007

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd). Springer. https://doi.org/10.1007/978-0-387-84858-7

Hernández, L., Baladron, C., Aguiar, J. M., Carro, B., & Sánchez-Esguevillas, A. (2014). Artificial neural networks for short-term load forecasting in microgrids environment. *Energy*, *75*, 252–264. https://doi.org/10.1016/j.energy.2014.07.065

Hersbach, H., Bell, B., Berrisford, P., et al. (2020). The ERA5 global reanalysis. *Quart. J. Roy. Meteorol. Soc.*, *146*(730), 1999–2049. https://doi.org/10.1002/qj.3803

Hong, T., & Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. *Int. J. Forecasting*, *32*(3), 914–938. https://doi.org/10.1016/j.ijforecast.2015.11.011

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd). OTexts. https://otexts.com/fpp3/

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *Int. J. Forecasting*, *22*(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

International Energy Agency. (2017). *Digitalization and energy* (tech. rep.). IEA. https://www.iea.org/reports/digitalization-and-energy

Keerthisinghe, C., Chapman, A. C., & Verbic, G. (2018). A fast technique for smart home management: ADP with temporal difference learning. *IEEE Trans. Smart Grid*, *9*(4), 3291–3303. https://doi.org/10.1109/TSG.2016.2629510

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Phys. Rev. E*, *69*(6), 066138. https://doi.org/10.1103/PhysRevE.69.066138

Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *J. Econometrics*, *54*(1-3), 159–178.

Little, R. J. A., & Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd). Wiley. https://doi.org/10.1002/9781119013563

Ljung, G. M., & Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika*, *65*(2), 297–303.

Luthander, R., Widén, J., Nilsson, D., & Palm, J. (2015). Photovoltaic self-consumption in buildings: A review. *Appl. Energy*, *142*, 80–94. https://doi.org/10.1016/j.apenergy.2014.12.028

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS ONE*, *13*(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

McKenna, E., Pless, J., & Darby, S. J. (2018). Solar photovoltaic self-consumption in the UK residential sector: New estimates from a smart grid demonstration project. *Energy Policy*, *118*, 482–491. https://doi.org/10.1016/j.enpol.2018.04.006

Nguyen, D. T., & Le, L. B. (2015). Risk-constrained profit maximization for microgrid aggregators with demand response. *IEEE Trans. Smart Grid*, *6*(1), 135–146. https://doi.org/10.1109/TSG.2014.2346024

Palensky, P., & Dietrich, D. (2011). Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Trans. Ind. Informat.*, *7*(3), 381–388. https://doi.org/10.1109/TII.2011.2158841

Ratnam, E. L., Weller, S. R., Kellett, C. M., & Murray, A. T. (2017). Residential load and rooftop PV generation: An australian distribution network dataset. *Int. J. Sustain. Energy*, *36*(8), 787–806. https://doi.org/10.1080/14786451.2015.1100196

Raza, M. Q., Nadarajah, M., & Ekanayake, C. (2016). On recent advances in PV output power forecast. *Sol. Energy*, *136*, 125–144. https://doi.org/10.1016/j.solener.2016.06.073

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Stat.*, *6*(2), 461–464.

Taylor, J. W. (2008). An evaluation of methods for very short-term load forecasting using minute-by-minute british data. *Int. J. Forecasting*, *24*(4), 645–658. https://doi.org/10.1016/j.ijforecast.2008.07.001

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, *17*(6), 520–525. https://doi.org/10.1093/bioinformatics/17.6.520

Tsui, K. M., & Chan, S. C. (2012). Demand response optimization for smart home scheduling under real-time pricing. *IEEE Trans. Smart Grid*, *3*(4), 1812–1821. https://doi.org/10.1109/TSG.2012.2218835

Tuballa, M. L., & Abundo, M. L. (2016). A review of the development of smart grid technologies. *Renewable Sustain. Energy Rev.*, *59*, 710–725. https://doi.org/10.1016/j.rser.2016.01.011

Wang, Y., Chen, Q., Hong, T., & Kang, C. (2019). Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Trans. Smart Grid*, *10*(3), 3125–3148. https://doi.org/10.1109/TSG.2018.2818167

World Energy Council. (2019). *World energy trilemma index 2019* (tech. rep.). World Energy Council. https://www.worldenergy.org/publications/entry/world-energy-trilemma-index-2019

Yang, D., Kleissl, J., Gueymard, C. A., Pedro, H. T. C., & Coimbra, C. F. M. (2018). History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining. *Sol. Energy*, *168*, 60–101. https://doi.org/10.1016/j.solener.2017.11.023

Zhou, B., Li, W., Chan, K. W., Cao, Y., Kuang, Y., Liu, X., & Wang, X. (2016). Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable Sustain. Energy Rev.*, *61*, 30–40. https://doi.org/10.1016/j.rser.2016.03.047

A Mathematical Formulations

A.1 SARIMA Model Specification

The Seasonal Autoregressive Integrated Moving Average model SARIMA$(p,d,q)(P,D,Q)_s$ is defined as:

$$\Phi_P(B^s)\phi_p(B)(1-B)^d(1-B^s)^D Y_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t \tag{5}$$

where:

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p \quad \text{(AR polynomial)} \tag{6}$$

$$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q \quad \text{(MA polynomial)} \tag{7}$$

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps} \quad \text{(Seasonal AR)} \tag{8}$$

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs} \quad \text{(Seasonal MA)} \tag{9}$$

B denotes the backshift operator ($B^k Y_t = Y_{t-k}$), $s = 24$ is the seasonal period (hourly data with daily seasonality), and $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ is white noise.

A.2 XGBoost Objective Function

The XGBoost algorithm minimizes the regularized objective:

$$\mathscr{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{10}$$

where l is a differentiable convex loss function (squared error for regression), K is the number of trees, and the regularization term is:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{11}$$

Here, T is the number of leaves, w_j are the leaf weights, γ controls tree complexity, and λ is the L2 regularization parameter (set to 1.0 in this study).

A.3 Linear Programming Formulation for Battery Optimization

The complete LP formulation for the 24-hour battery dispatch problem is:

Objective: Minimize daily electricity cost

$$\min_{P^{\text{imp}}, P^{\text{exp}}, P^{\text{ch}}, P^{\text{dis}}} \sum_{t=1}^{24} \left(\pi_t^{\text{buy}} \cdot P_t^{\text{imp}} - \pi_t^{\text{sell}} \cdot P_t^{\text{exp}} \right) \cdot \Delta t \tag{12}$$

Subject to:

Power Balance:

$$P_t^{\text{PV}} + P_t^{\text{imp}} + P_t^{\text{dis}} = D_t + P_t^{\text{exp}} + P_t^{\text{ch}} \quad \forall t \in \{1, \ldots, 24\} \tag{13}$$

State of Charge Dynamics:

$$\text{SOC}_{t+1} = \text{SOC}_t + \eta^{\text{ch}} P_t^{\text{ch}} \cdot \Delta t - \frac{P_t^{\text{dis}}}{\eta^{\text{dis}}} \cdot \Delta t \tag{14}$$

Capacity Constraints:

$$0 \leq \text{SOC}_t \leq E^{\max} = 10 \text{ kWh} \tag{15}$$
$$0 \leq P_t^{\text{ch}} \leq P^{\text{ch,max}} = 5 \text{ kW} \tag{16}$$
$$0 \leq P_t^{\text{dis}} \leq P^{\text{dis,max}} = 5 \text{ kW} \tag{17}$$
$$0 \leq P_t^{\text{imp}} \leq P^{\text{grid,max}} = 5 \text{ kW} \tag{18}$$
$$0 \leq P_t^{\text{exp}} \leq P^{\text{grid,max}} = 5 \text{ kW} \tag{19}$$

Boundary Conditions:

$$\text{SOC}_1 = 0.5 \cdot E^{\max} = 5 \text{ kWh} \quad \text{(initial)} \tag{20}$$
$$\text{SOC}_{25} \geq 0.2 \cdot E^{\max} = 2 \text{ kWh} \quad \text{(final)} \tag{21}$$

where $\eta^{\text{ch}} = \eta^{\text{dis}} = \sqrt{0.95} \approx 0.975$ (round-trip efficiency 95%), $\Delta t = 1$ hour, and $\pi_t^{\text{sell}} = 0.5 \cdot \pi_t^{\text{buy}}$ (feed-in tariff at 50% of retail price).

B Evaluation Metrics

B.1 Error Metrics Definitions

The following metrics were used throughout this study:

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{22}$$

Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{23}$$

Normalized RMSE (nRMSE):

$$\text{nRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}} \tag{24}$$

Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{25}$$

Note: MAPE was not used as the primary metric due to its sensitivity to near-zero demand values, which are common during nighttime hours.

B.2 Statistical Tests

Augmented Dickey-Fuller (ADF) Test: Tests for unit root in the time series. The test statistic is:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \varepsilon_t \tag{26}$$

H_0: $\gamma = 0$ (unit root exists, series is non-stationary).

KPSS Test: Tests for stationarity around a deterministic trend:

$$y_t = \xi t + r_t + \varepsilon_t, \quad r_t = r_{t-1} + u_t \tag{27}$$

H_0: $\sigma_u^2 = 0$ (series is trend-stationary).

C Data Processing Details

C.1 Feature Engineering Summary

Table 13 provides a complete list of engineered features used in the XGBoost model.

Table 13: Complete feature set for XGBoost demand forecasting.

Category	Feature	Description
3*Temporal	hour_sin	$\sin(2\pi \cdot \text{hour}/24)$
	hour_cos	$\cos(2\pi \cdot \text{hour}/24)$
	is_weekend	Binary: 1 if Saturday/Sunday
5*Autoregressive	demand_lag_1	Demand at $t-1$ hour
	demand_lag_2	Demand at $t-2$ hours
	demand_lag_3	Demand at $t-3$ hours
	demand_lag_24	Demand at $t-24$ hours (same hour yesterday)
	demand_lag_168	Demand at $t-168$ hours (same hour last week)
5*Exogenous	temperature	Ambient temperature (°C)
	shortwave_radiation	Solar irradiance (W/m^2)
	cloud_cover	Cloud cover fraction (%)
	price	Electricity price (/kWh)
	HDD/CDD	Heating/Cooling degree days

C.2 Missing Data Statistics

Table 14 summarizes the missing data patterns identified in the raw dataset.

Table 14: Missing value summary for raw sensor data.

Variable	Total Obs.	Missing	% Missing
Demand	8,760	0	0.00%
pv_mod1	8,760	247	2.82%
pv_mod2	8,760	89	1.02%
pv_mod3	8,760	112	1.28%
Temperature	8,760	12	0.14%
Price	8,760	0	0.00%

D Computational Environment

D.1 Hardware and Software

All experiments were conducted on a MacBook Air (M1, 2020) with 8 GB RAM. The software environment consisted of:

- Python 3.10.12
- pandas 2.1.4, numpy 1.26.2

- xgboost 2.0.3, scikit-learn 1.3.2
- statsmodels 0.14.1
- cvxpy 1.4.1 with GLPK solver
- matplotlib 3.8.2, seaborn 0.13.0, plotly 5.18.0

D.2 Reproducibility

All random operations were seeded with `RANDOM_SEED = 42` to ensure reproducibility. The complete codebase is available at:

https://github.com/samuel29102002/Energy_Data_Science

E Additional Model Diagnostics

E.1 Residual Analysis

For the selected SARIMA(1,1,1)(1,1,1,24) model, the Ljung-Box test (Ljung & Box, 1978) on residuals yielded $Q(24) = 28.4$ with $p = 0.24$, indicating no significant autocorrelation remaining in the residuals. The Jarque-Bera test for normality gave $JB = 142.3$ with $p < 0.001$, suggesting non-normal residuals with excess kurtosis—likely due to occasional demand spikes not captured by the model.

E.2 Model Selection Criteria

Table 15 presents the information criteria for candidate ARIMA models (Akaike, 1974; Schwarz, 1978). Lower values indicate better fit-complexity trade-offs.

Table 15: Information criteria for ARIMA model selection.

Model	AIC	BIC	Log-Likelihood
ARIMA(2,1,2)	-4,521.3	-4,489.6	2,266.7
SARIMA(1,1,1)(1,1,1,24)	-5,892.1	-5,847.2	2,953.1
SARIMA(2,1,1)(0,1,1,24)	-5,876.4	-5,838.0	2,944.2

The SARIMA(1,1,1)(1,1,1,24) model achieves the lowest AIC and BIC, confirming its selection as the preferred statistical model.

E.3 XGBoost Learning Curves

The XGBoost model was trained with early stopping disabled to ensure consistent iteration counts across experiments. The training RMSE decreased monotonically from 0.42 (iteration 1) to 0.18 (iteration 600), while validation RMSE stabilized around 0.29 after approximately 400 iterations, indicating mild overfitting in the final 200 iterations. Future work could employ early stopping with patience=50 to reduce training time.

F Sensitivity Analysis

F.1 Battery Capacity Sensitivity

To understand the marginal value of battery storage, I conducted a sensitivity analysis varying battery capacity from 0 to 20 kWh. Results for the PV_high scenario:

Table 16: Daily cost sensitivity to battery capacity (PV_high scenario).

Capacity (kWh)	Daily Cost ()	Marginal Value (/kWh)
0	0.52	–
5	0.18	0.068
10	-0.05	0.046
15	-0.12	0.014
20	-0.14	0.004

The marginal value of additional storage decreases rapidly beyond 10 kWh, suggesting diminishing returns for larger batteries in this household context.

F.2 Price Volatility Impact

The optimization benefits increase with price volatility. Under the observed price spread (max/min ratio ≈ 3.2), the battery achieves meaningful arbitrage. Simulations with reduced price spread (ratio = 1.5) showed the PV_high scenario cost increasing from -0.05 to +0.08, eliminating the profit margin.

G Limitations and Future Work

G.1 Known Limitations

1. **Perfect foresight assumption:** The optimization assumes perfect knowledge of future prices and PV generation, which is unrealistic in practice. Model Predictive Control (MPC) with rolling updates would be more realistic (Keerthisinghe et al., 2018).
2. **Single household:** Results may not generalize to different household types (e.g., families vs. single occupants, heat pumps vs. gas heating) (Ratnam et al., 2017).
3. **Static battery model:** Battery degradation, temperature effects, and state-of-health variations were not modeled (Erdinc & Uzunoglu, 2012).
4. **No demand response:** The model treats demand as inelastic; incorporating flexible loads (EV charging, water heaters) could unlock additional value (Nguyen & Le, 2015).

G.2 Extensions for Future Work

- Implement stochastic optimization with scenario-based uncertainty quantification

- Add EV charging as a flexible load with departure time constraints
- Compare deep learning approaches (LSTM, Transformer) for demand forecasting
- Develop a real-time MPC controller with rolling horizon optimization
- Include carbon intensity signals for environmental optimization