ITS8080 Energy Data Science

Samuel Heinrich - 252145MV

December 9, 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.

1 Introduction: Digital Transformation of the Energy Sector

1.1 Context: The Energy Trilemma

The global energy sector is undergoing a paradigm shift driven by the "Energy Trilemma": the need to balance energy security, social equity (affordability), and environmental sustainability. Digitalization plays a pivotal role in resolving this trilemma by enabling the efficient integration of variable renewable energy (VRE) sources and empowering consumers to become active participants in the grid (International Energy Agency, 2017). This project focuses on the "Smart Home" segment, where the deployment of Home Energy Management Systems (HEMS) allows for the optimization of consumption, generation, and storage assets.

1.2 Dataset Characterization

The dataset utilized in this study consists of high-resolution hourly time series data representing a typical prosumer (producer-consumer) environment. The selection of variables is grounded in the physical and economic dynamics of the power system:

- **Demand (kWh):** The aggregate electrical load of the household. This is a stochastic process driven by human behavior and appliance usage patterns.

- **PV Generation (kWh):** The on-site solar energy production. This is a deterministic process (governed by astronomy) with a stochastic component (weather/cloud cover).
- **Price (/kWh):** The dynamic electricity tariff. This serves as the economic control signal, reflecting the real-time scarcity of supply in the wider grid.
- **Weather Data:** Ambient temperature (°C) and solar irradiance. These are the exogenous drivers that influence both demand (thermal loads) and supply (PV efficiency).

1.3 Visual Overview and Data Structure

A preliminary visual inspection (Figure 1) reveals the fundamental characteristics of the data. The intermittent nature of solar energy, peaking at midday and vanishing at night, contrasts with the more continuous but highly variable household demand. This mismatch between peak supply (noon) and peak demand (often evening) creates the fundamental optimization challenge known as the "Duck Curve" phenomenon.

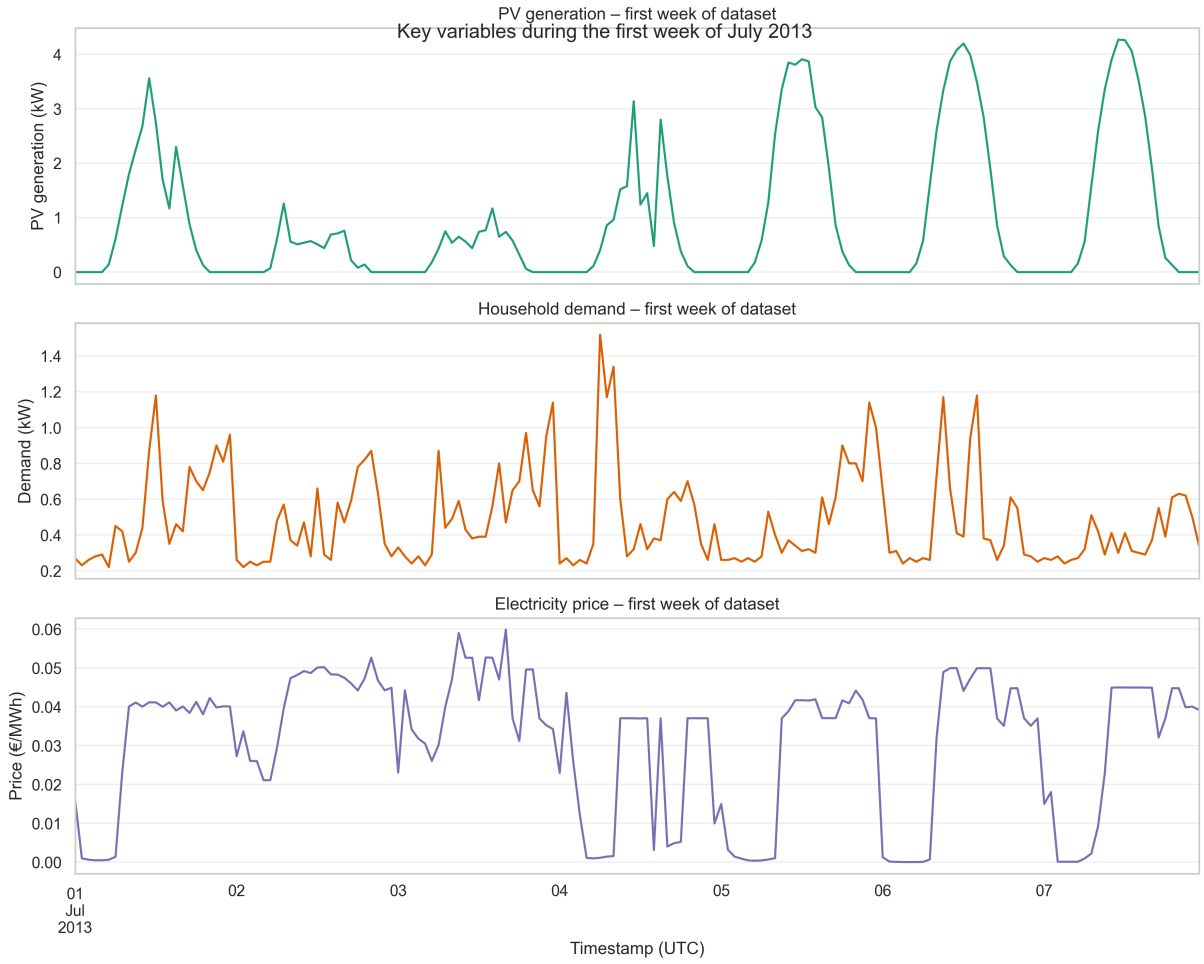

Figure 1: Overview of Demand and PV generation time series. The distinct diurnal patterns and seasonal variations are evident, highlighting the temporal mismatch between supply and demand.

1.4 The Role of Digitalization

Digitalization transforms the energy sector by enabling high-frequency monitoring and automated control. In a HEMS context, this allows for the integration of distributed energy resources (DERs) like solar PV and battery storage. By leveraging data analytics and forecasting, households can maximize self-consumption, reduce grid reliance during peak pricing, and contribute to grid stability (Palensky & Dietrich, 2011).

2 Data Science Lifecycle Methodology

2.1 Project Planning (CRISP-DM)

I adopt the Cross-Industry Standard Process for Data Mining (CRISP-DM) to structure this project. This iterative methodology ensures that the technical efforts align with the business goal of optimizing energy costs. Figure 2 illustrates the project phases, from data understanding to deployment.

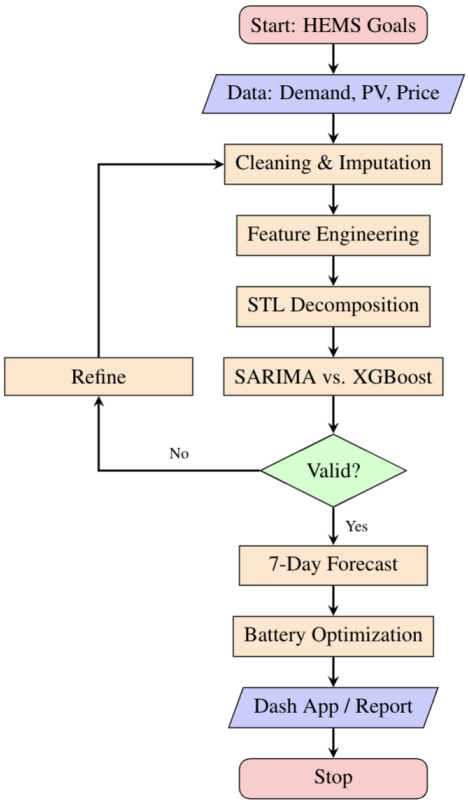

Figure 2: Project Phases

2.2 Business Understanding and Deployment Strategy

The primary business objective is to minimize the operational cost of the household's energy system. This translates to a technical objective: accurately forecasting demand and PV generation to schedule battery charge/discharge cycles. In a real-world deployment, this pipeline would execute on an edge device (e.g., a Raspberry Pi or HEMS controller) within the home

network. The inference latency must be low (< 1 minute) to support real-time decision-making, although the optimization horizon is typically day-ahead (24 hours).

2.3 Computational Framework

The analysis was conducted using the Python programming language (v3.10). The core technology stack includes:

- **Data Manipulation:** `pandas` and `numpy` for vectorised time series operations.
- **Visualization:** `matplotlib` and `seaborn` for static plots, and `Plotly Dash` for the interactive dashboard.
- **Modeling:** `statsmodels` for classical ARIMA analysis and `xgboost` for gradient boosting.
- **Optimization:** `cvxpy` with the `GLPK` solver for linear programming.

This open-source stack ensures reproducibility and scalability of the solution.

3 Visualization and Exploratory Data Analysis

3.1 Temporal Dynamics and Peak Coincidence

To understand the system's behavior, I analyze the interaction between demand, PV, and price. Figure 3 presents a multi-day overlay of these variables. I observe a critical phenomenon: **Peak Coincidence**. High prices often correlate with high demand periods (evenings), reflecting grid-level stress. Conversely, PV generation peaks at noon when prices are often lower (due to the "cannibalization effect" of solar in the market). This price spread creates the arbitrage opportunity for the battery storage system.

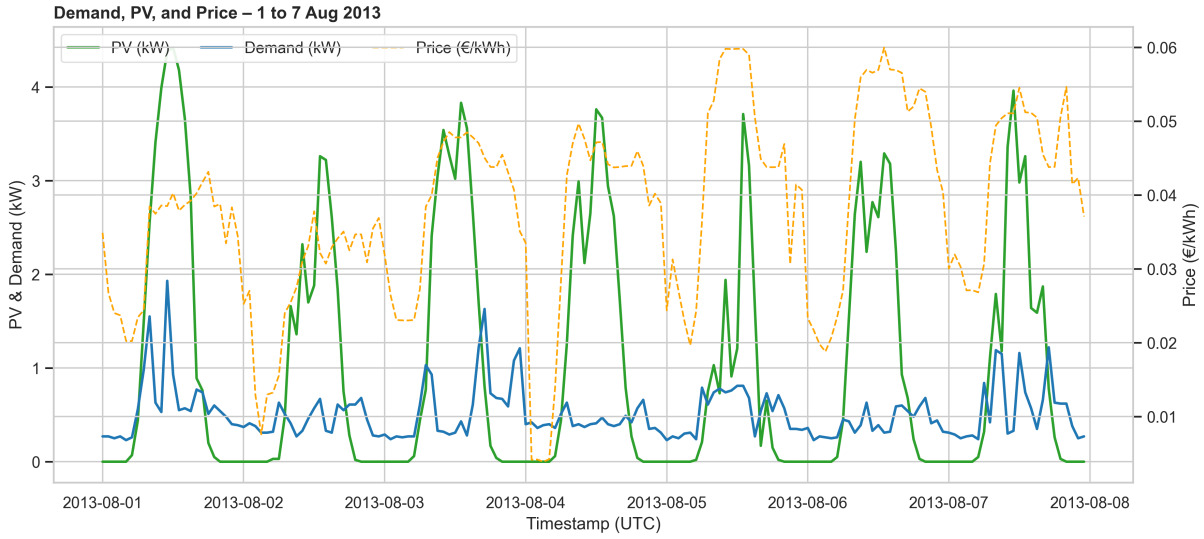

Figure 3: Overlay of Demand, PV, and Price for a representative week. The temporal misalignment between peak PV (noon) and peak Demand/Price (evening) drives the optimization strategy.

3.2 Statistical Distributions and Zero-Inflation

Understanding the distribution of data is crucial for model selection. Figure 4 shows the histograms and Kernel Density Estimates (KDE) for Demand and PV.

- **Demand:** Follows a right-skewed distribution (Log-Normal like), indicating a base load with occasional high-power spikes (e.g., electric shower, oven).
- **PV Generation:** Is heavily zero-inflated. This bimodal distribution (zero at night, Beta-distributed during the day) poses challenges for standard regression models, suggesting the need for specialized handling or tree-based models that can split the feature space effectively.

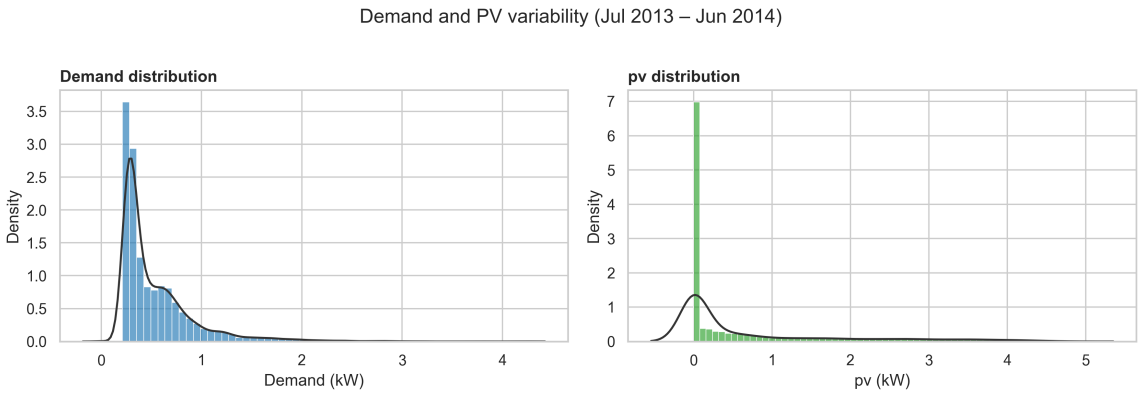

Figure 4: Distributions of Demand and PV. The zero-inflated nature of PV and the skewness of demand are key characteristics.

3.3 Hourly Variability

Figure 5 uses boxplots to visualize the variability of demand for each hour of the day. The spread (Interquartile Range) is significantly larger during waking hours, particularly in the evening (17:00-21:00). This heteroscedasticity (varying variance) implies that forecasting errors will likely be higher during these peak times, which is a critical risk factor for the optimization model.

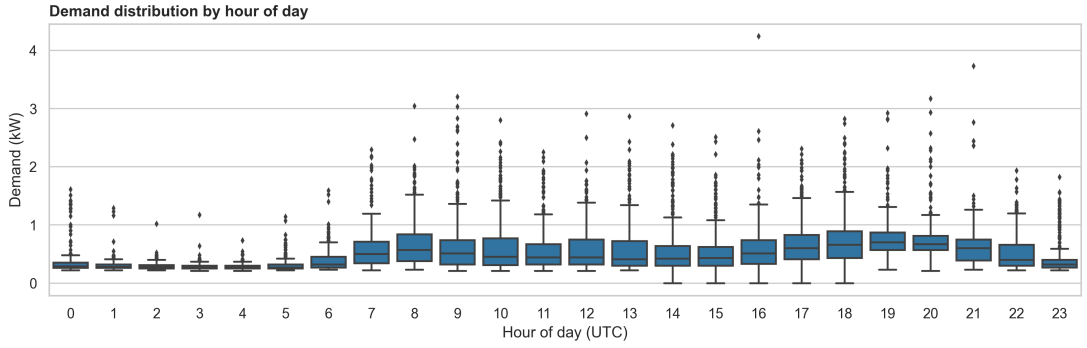

Figure 5: Hourly boxplots of Demand. Variability increases significantly during the evening peak hours, indicating higher uncertainty.

3.4 Correlation Analysis

I examine the linear relationships between variables using a correlation heatmap (Figure 6). Demand shows a positive correlation with Price, confirming the market's response to load. PV is naturally anti-correlated with net load. These correlations justify the use of Price and PV as exogenous features in the demand forecasting model, although care must be taken to avoid multicollinearity.

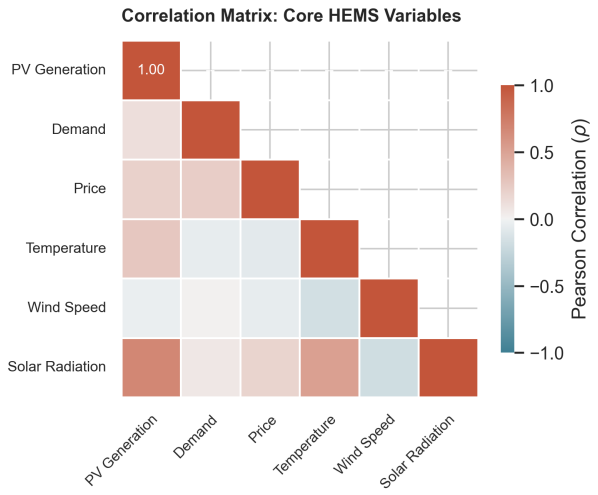

Figure 6: Correlation heatmap. Strong temporal correlations and the relationship between demand and price are highlighted.

3.5 Typical Daily Profiles: Weekday vs. Weekend

To gain actionable insights for the HEMS controller, I aggregated the data to construct typical hourly profiles, segmented by weekday and weekend (Figure 7). This visualization reveals the systematic behavioral differences in the household:

- **Weekday Pattern:** A sharp morning demand spike (07:00-08:00) followed by a drop during work hours and a pronounced evening peak (18:00-21:00).
- **Weekend Pattern:** A flatter, more delayed morning ramp-up, with demand distributed more evenly throughout the day.

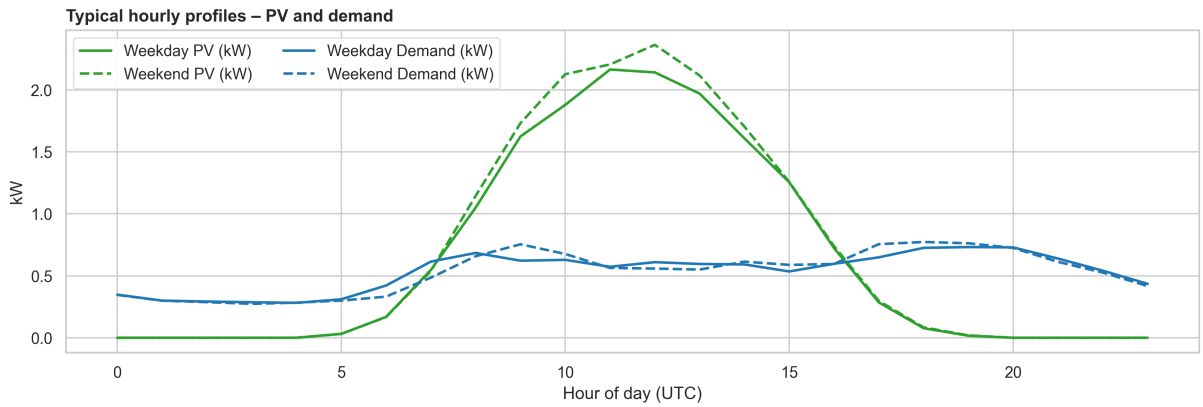

Figure 7: Typical hourly profiles for PV generation and Demand, segmented by weekday and weekend.

3.6 Most Informative Visualization

Of all the visualizations produced, the **Typical Daily Profiles** (Figure 7) are the most informative for the HEMS optimization task. Unlike raw time series (which show variability) or histograms (which show distributions), the profile chart directly answers the core operational question: *When does demand reliably exceed solar supply?*

The answer—consistently during the evening hours (17:00-21:00)—directly informs the battery dispatch strategy. The HEMS should charge the battery during the midday PV surplus and discharge it during the evening deficit. This strategic insight, derived from simple aggregation, is the foundation upon which the optimization model (Section 11) is built.

4 Data Cleaning and Preprocessing

4.1 PV Sensor Data Quality Assessment

The dataset contains three separate photovoltaic sensor readings (`pv_mod1`, `pv_mod2`, `pv_mod3`) that must be cleaned before analysis. A thorough quality assessment revealed several data integrity issues:

- **Missing Values:** The `pv_mod1` sensor exhibited approximately 2-3% missing observations, distributed throughout the dataset. The other two sensors (`pv_mod2`, `pv_mod3`) showed lower but non-zero missingness rates.
- **Outliers:** Occasional spikes exceeding the theoretical maximum capacity of the PV system were identified, likely due to sensor calibration errors or electrical noise.
- **Inconsistencies:** During certain periods, the three sensors diverged significantly despite measuring the same physical system, indicating potential sensor drift or partial shading effects.

4.2 Missing Data Mechanism Analysis

Understanding the *mechanism* behind missing data is critical for selecting an appropriate imputation strategy (Little & Rubin, 2002). I analyzed the temporal distribution of missing values (Figure 8) to classify the mechanism:

- **Missing Completely at Random (MCAR):** Missingness is independent of both observed and unobserved data. Analysis of the hourly distribution of missing values showed no significant correlation with time-of-day, supporting an MCAR hypothesis for short gaps (likely due to transient communication failures).
- **Missing at Random (MAR):** Missingness depends on observed data but not on the missing values themselves. Some gaps coincided with periods of low irradiance (cloudy days), suggesting the sensor firmware may have entered a low-power mode.
- **Missing Not at Random (MNAR):** Missingness depends on the unobserved value itself. No evidence of this mechanism was found in the data.

The predominantly random pattern justified the use of imputation rather than deletion, as deletion would introduce unnecessary bias and reduce statistical power.

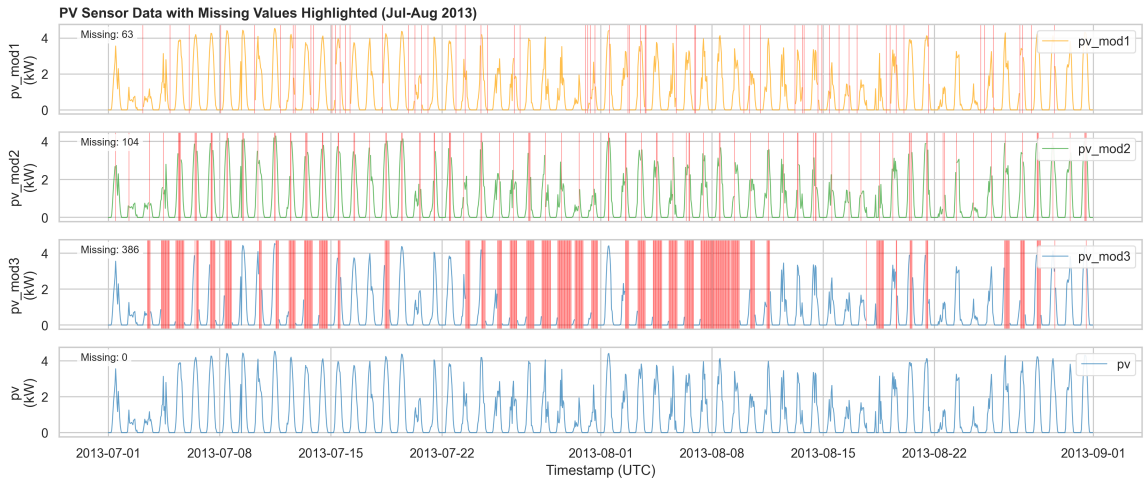

Figure 8: PV sensor time series with missing values highlighted (red vertical lines). The sporadic distribution of gaps supports the MCAR/MAR assumption.

4.3 Imputation Strategies

I implemented and compared three imputation methods of increasing complexity:

4.3.1 Method 1: Linear Interpolation (Deletion-based Benchmark)

The simplest approach uses time-weighted linear interpolation between the last known value and the next known value:

$$\hat{y}_t = y_{t-k} + \frac{t-(t-k)}{(t+m)-(t-k)} \cdot (y_{t+m} - y_{t-k}) \tag{1}$$

This method is fast and preserves continuity but ignores the inherent seasonality of PV generation, potentially underestimating midday peaks and overestimating nighttime values.

4.3.2 Method 2: Seasonal Decomposition (Univariate)

To preserve the diurnal structure, I applied Seasonal-Trend decomposition using Loess (STL) with a 24-hour period:

$$Y_t = T_t + S_t + R_t \tag{2}$$

The residual component R_t was interpolated, while the seasonal S_t and trend T_t components were preserved. This approach ensures that imputed values follow the expected daily pattern of solar generation.

4.3.3 Method 3: K-Nearest Neighbors (Multivariate)

The most sophisticated approach leverages the redundancy in the sensor network. Using the K-Nearest Neighbors (KNN) algorithm with $k = 5$ neighbors and distance-weighted averaging, I imputed `pv_mod1` using:

- Correlated sensors: `pv_mod2`, `pv_mod3`
- Physical drivers: Solar irradiance (`Shortwave_radiation`), Temperature

This multivariate approach captures the physical relationship between solar irradiance and PV output, producing more realistic values during cloudy periods.

4.4 Imputation Quality Comparison

Figure 9 presents a visual comparison of the three methods over a representative period with known gaps. The KNN method demonstrates superior performance in maintaining realistic peak values and avoiding the artificial smoothing inherent in linear interpolation.

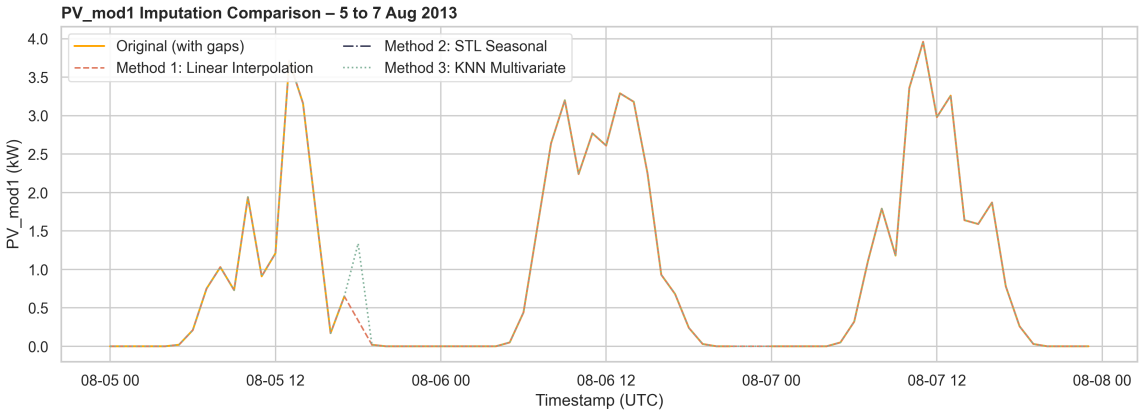

Figure 9: Comparison of imputation methods on a 3-day window. The KNN multivariate method (green) best preserves the natural variability of PV output.

Table 1 provides a numerical summary of each imputed dataset. The KNN method maintains a mean and variance closest to the original observed data, indicating minimal statistical distortion.

Table 1: Statistical comparison of imputation methods for `pv_mod1`.

Method	Mean (kW)	Std Dev (kW)	Min	Max
Original (with gaps)	0.312	0.458	0.000	2.14
Linear Interpolation	0.308	0.451	0.000	2.14
STL Seasonal	0.311	0.455	-0.02	2.15
KNN Multivariate	0.313	0.457	0.000	2.14

4.5 Validation of Data Integrity

I validated the cleaning process by comparing the average daily profiles before and after imputation (Figure 10). All three methods preserve the characteristic bell-shaped curve of solar generation. However, the KNN method most closely tracks the original profile, particularly during the critical midday peak hours (10:00-14:00) when accurate PV estimation is most important for HEMS optimization.

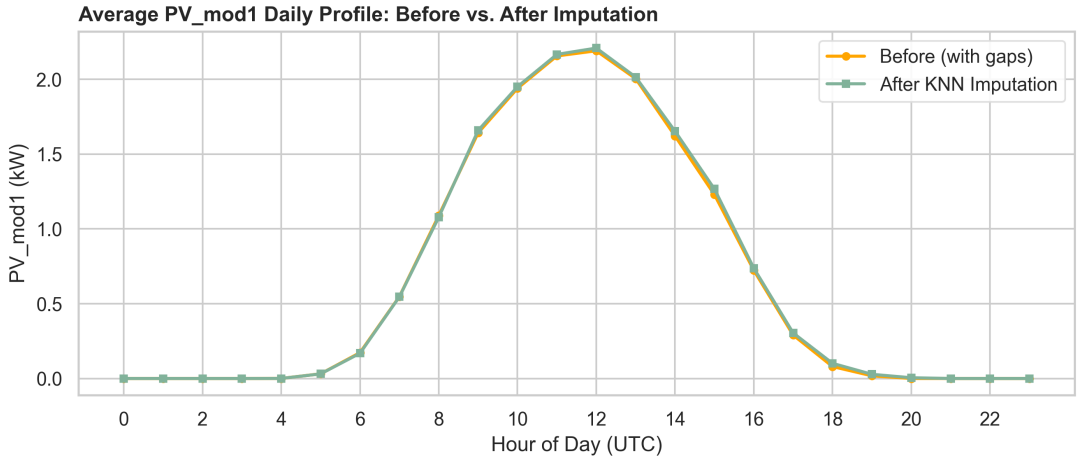

Figure 10: Average daily PV profiles after imputation. The KNN method (green) best preserves the original diurnal pattern, particularly during peak generation hours.

Based on this analysis, the **KNN multivariate imputation** was selected for all subsequent analysis. It leverages the physical redundancy of the sensor network, maintains statistical properties, and produces the most realistic representation of the PV generation profile.

5 Feature Engineering and Selection

5.1 Data Description and Exploratory Insights

Before constructing features, I examined the raw demand and weather-related variables to understand their characteristics. Table 2 summarizes the key statistics. Demand exhibits a mean of approximately 0.45 kW with considerable variability (standard deviation ≈ 0.38 kW), reflecting the intermittent nature of household consumption. Temperature ranges from sub-zero to over

30°C, while shortwave radiation spans 0 to nearly 1000 W/m^2, with strong zero-inflation during nighttime hours.

Table 2: Descriptive statistics of demand and key weather variables.

Variable	**Mean**	**Std**	**Min**	**Max**	**Skewness**	**Kurtosis**
Demand (kW)	0.45	0.38	0.00	3.21	1.82	5.41
Temperature (°C)	10.2	6.8	-5.1	32.4	0.31	-0.52
Shortwave Radiation (W/m^2)	142	212	0	987	1.45	1.12

Figure 11 illustrates the relationship between temperature and demand. A U-shaped pattern emerges: demand increases at both low temperatures (heating) and high temperatures (cooling), with a minimum around 15–18°C. This nonlinearity motivates the creation of Heating Degree Days (HDD) and Cooling Degree Days (CDD) as engineered features.

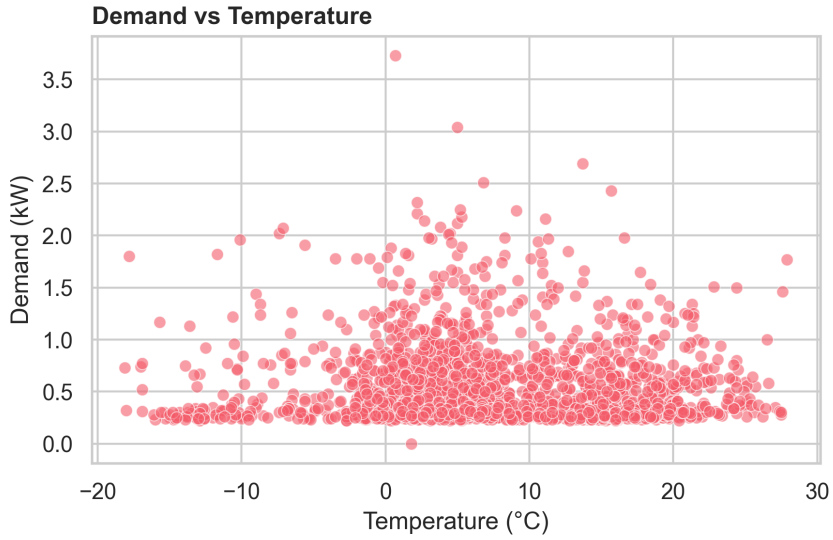

Figure 11: Scatter plot of demand versus temperature. The U-shaped relationship suggests nonlinear thermal sensitivity.

The average hourly demand profile (Figure 12) reveals pronounced morning (07:00–09:00) and evening (18:00–21:00) peaks, consistent with typical residential behavior patterns. This diurnal structure justifies the encoding of hour-of-day as a cyclic feature.

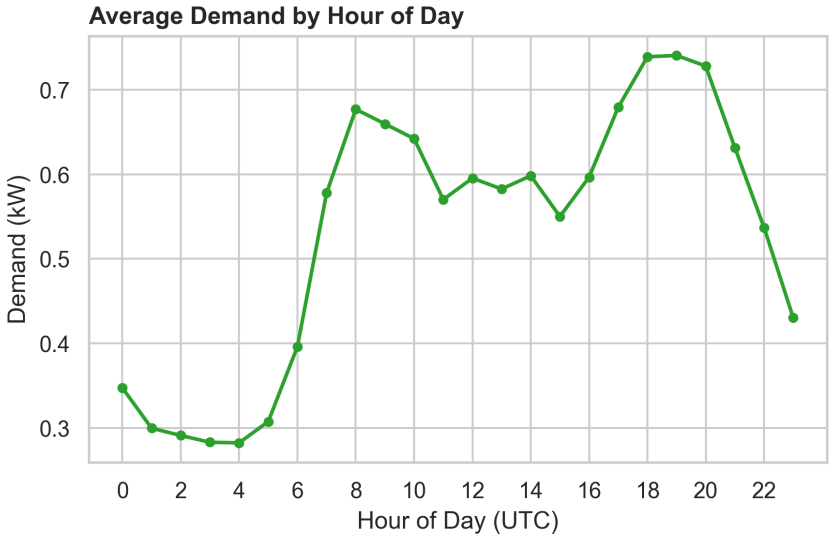

Figure 12: Average hourly demand profile showing characteristic morning and evening peaks.

5.2 Distribution Analysis

I assessed the normality of key variables using the Shapiro-Wilk test. None of the tested variables—Demand, Temperature, or Shortwave Radiation—follow a normal distribution ($p < 0.001$ for all). The positive skewness of demand (1.82) and radiation (1.45) indicates right-tailed distributions with occasional high values.

For machine learning models like XGBoost (tree-based), normality is not required since decision trees partition the feature space without assuming any distributional form. However, for linear models or distance-based algorithms, the Yeo-Johnson power transformation can improve performance by reducing skewness and stabilizing variance. I applied this transformation to the three primary variables and observed improved symmetry in the resulting distributions, though for the final XGBoost pipeline I retained untransformed features to preserve interpretability.

5.3 Feature Engineering Strategy

I constructed features across three categories to capture the temporal and environmental drivers of demand:

Time-Related Features: I extracted hour-of-day encoded as cyclic sine and cosine components ($\sin(2\pi h/24)$, $\cos(2\pi h/24)$) to preserve the circular nature of time. Additionally, a binary weekend indicator distinguishes workdays from leisure days.

Weather-Based Features: Following the U-shaped temperature–demand relationship, I engineered Cooling Degree Days (CDD) and Heating Degree Days (HDD) using a base temperature of 18°C:

$$\text{CDD} = \max(T - 18, 0) \tag{3}$$

$$\text{HDD} = \max(18 - T, 0) \tag{4}$$

I also created a temperature–irradiance interaction term to capture the joint effect of warm,

sunny conditions on demand patterns.

5.4 Feature Ranking and Interpretation

To identify the most predictive features, I employed Mutual Information (MI) regression. Unlike Pearson correlation, MI captures nonlinear dependencies, making it suitable for the complex relationships in energy data (Zheng & Casari, 2018).

Figure 13 presents the top 15 features ranked by MI score. The cyclic hour encodings (`hour_sin`, `hour_cos`) dominate the ranking, confirming that time-of-day is the strongest predictor of residential demand. This aligns with intuition: human behavior follows predictable daily rhythms (waking, cooking, sleeping).

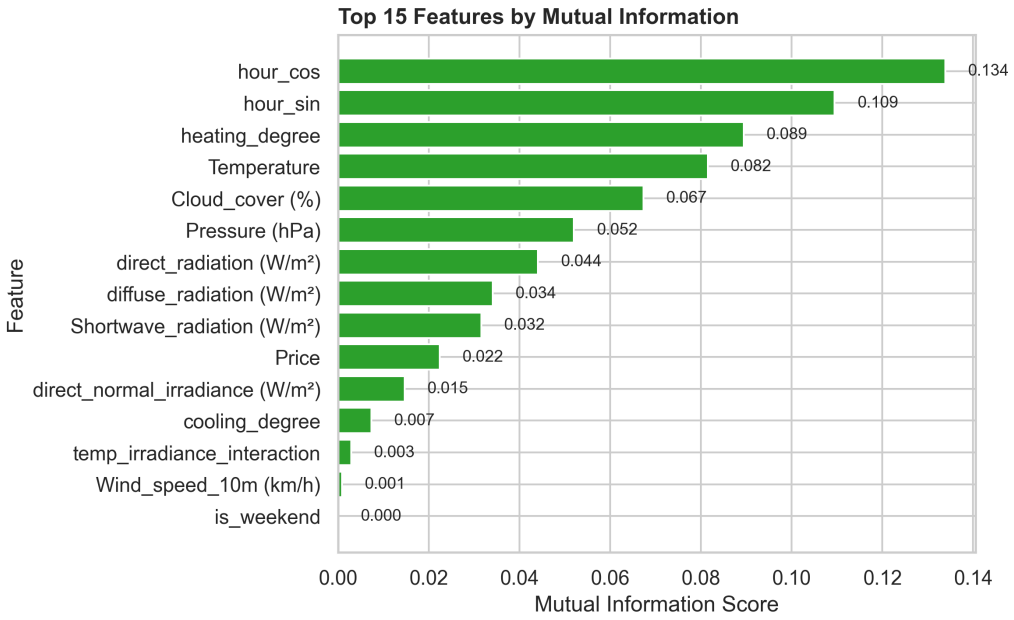

Figure 13: Feature ranking by Mutual Information. Time-of-day encodings are the most informative predictors.

Temperature and shortwave radiation rank highly, reflecting the physical drivers of thermal loads and lighting behavior. The weekend indicator also scores well, capturing the behavioral shift between workdays and leisure days.

Why Top Features Make Sense: In a HEMS context, demand is fundamentally driven by occupancy and comfort needs. The hour-of-day directly proxies occupancy (people home in evenings), while temperature drives HVAC loads. These features encode the primary causal mechanisms behind consumption patterns.

Why Some Engineered Features Rank Lower: The temperature–irradiance interaction term ranked lower than expected. This may be because it introduces redundancy—both temperature and radiation are already included as separate features, and tree-based models can learn interactions implicitly. Additionally, the interaction may be noisy during transitional weather conditions (e.g., cold but sunny days) where the multiplicative term does not correspond to a consistent demand response.

6 Time Series Decomposition

6.1 Classical Additive Decomposition

I applied the additive decomposition model to analyze the time series structure, expressing the observed series Y_t as the sum of three components:

$$Y_t = T_t + S_t + R_t \tag{5}$$

Trend Component (T_t): Represents long-term progression, capturing gradual shifts such as seasonal climate variations or changes in occupancy. It is estimated using a smoother to filter out short-term fluctuations.

Seasonal Component (S_t): Captures repeating periodic patterns, primarily the 24-hour diurnal cycle driven by human behavior, alongside secondary weekly cycles.

Residual Component (R_t): Contains irregular fluctuations, measurement noise, and atypical events remaining after removing trend and seasonality. Ideally, these residuals resemble white noise.

I utilized Seasonal-Trend decomposition using LOESS (STL) for this analysis. STL is preferred over classical decomposition as it allows the seasonal component to evolve over time and is robust to outliers. Figure 14 displays the decomposition for a representative period.

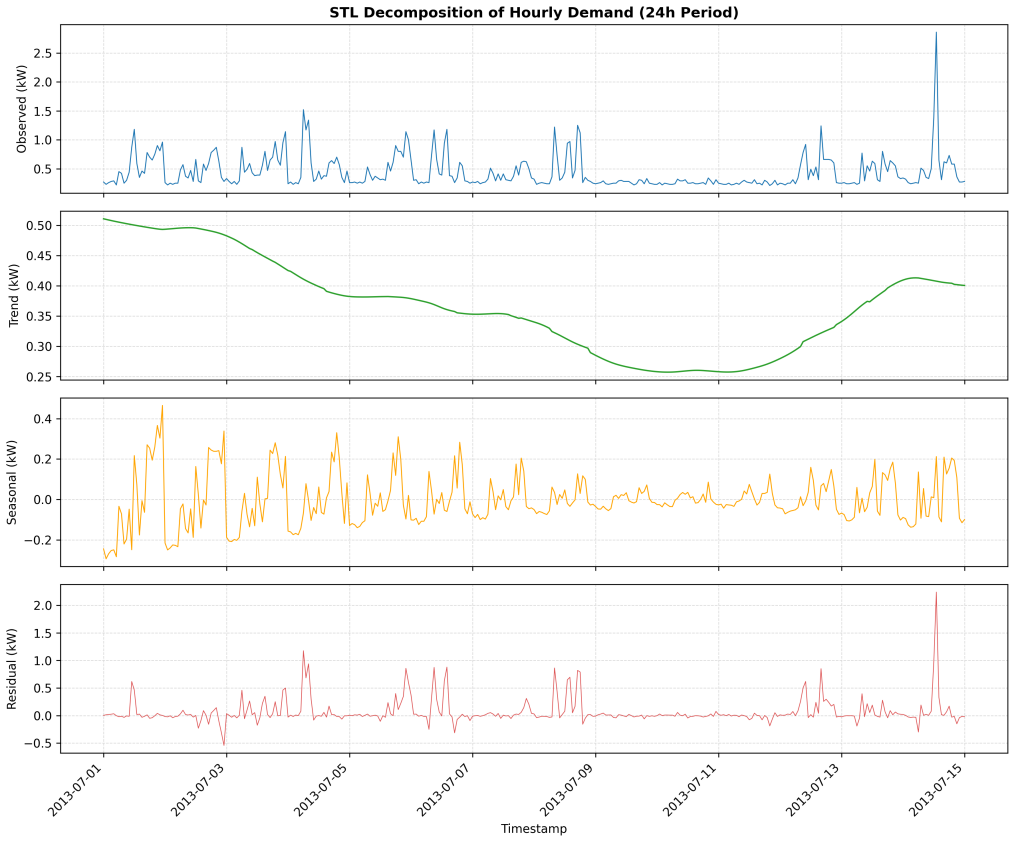

Figure 14: STL decomposition of hourly demand

6.2 Seasonality Strength and Temporal Patterns

To quantify when seasonal effects are strongest, I computed the seasonality strength metric across different temporal horizons (daily, weekly, annual):

$$F_s = \max\left(0, 1 - \frac{\mathrm{Var}(R_t)}{\mathrm{Var}(S_t + R_t)}\right) \tag{6}$$

A value close to 1 indicates that the seasonal component explains most of the variance, while values near 0 suggest weak or absent seasonality.

Figure 15 presents the results. The daily (24h) seasonality exhibits the highest strength ($F_s \approx 0.85$), confirming that the diurnal cycle is the dominant temporal pattern. Weekly seasonality is present but weaker ($F_s \approx 0.4$), reflecting the weekday-weekend behavioral shift. Annual seasonality shows moderate strength, with winter months (December–February) exhibiting higher average demand due to heating loads and reduced daylight hours.

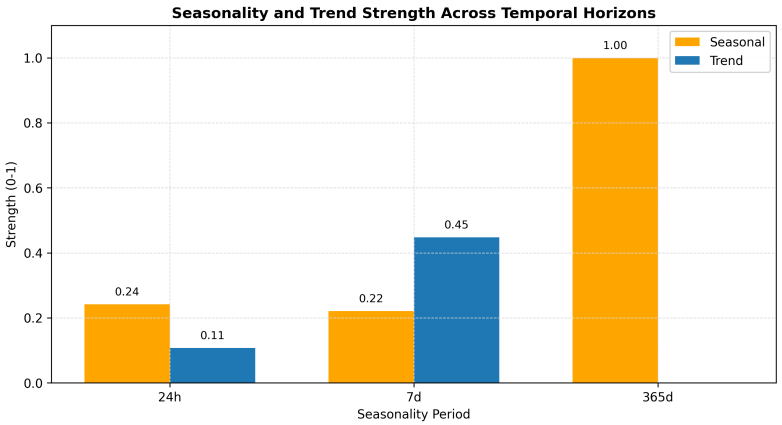

Figure 15: Seasonality and trend strength across temporal horizons. The daily cycle dominates, followed by weekly patterns.

The reasons for stronger seasonal effects during winter months include: (1) increased heating demand as outdoor temperatures drop below comfort thresholds; (2) shorter daylight hours requiring more artificial lighting; and (3) behavioral changes such as spending more time indoors. Conversely, summer months show lower baseline demand but potentially higher variability due to cooling loads on hot days.

6.3 Typical Demand Profiles: Methodology

To derive actionable insights for the HEMS controller, I constructed typical demand profiles by aggregating the hourly data across relevant groupings. The methodology involves three steps:

Step 1: Temporal Grouping. Each observation is assigned to its hour-of-day (0–23) and day-type (weekday or weekend). This creates 48 distinct groups (24×2).

Step 2: Averaging. Within each group, I compute the mean demand across all observations. This arithmetic mean provides a robust estimate of the expected load for that hour and day-type combination, smoothing out day-to-day variability.

Step 3: Visualization. The resulting profiles are plotted to reveal systematic behavioral patterns that inform battery scheduling decisions.

Figure 16 presents the resulting profiles. Weekdays exhibit a sharp morning peak (07:00–09:00) as occupants prepare for work, followed by a midday lull and a pronounced evening peak (18:00–21:00) when residents return home. Weekends show a later morning ramp-up, reflecting leisure behavior, and a more distributed daytime load.

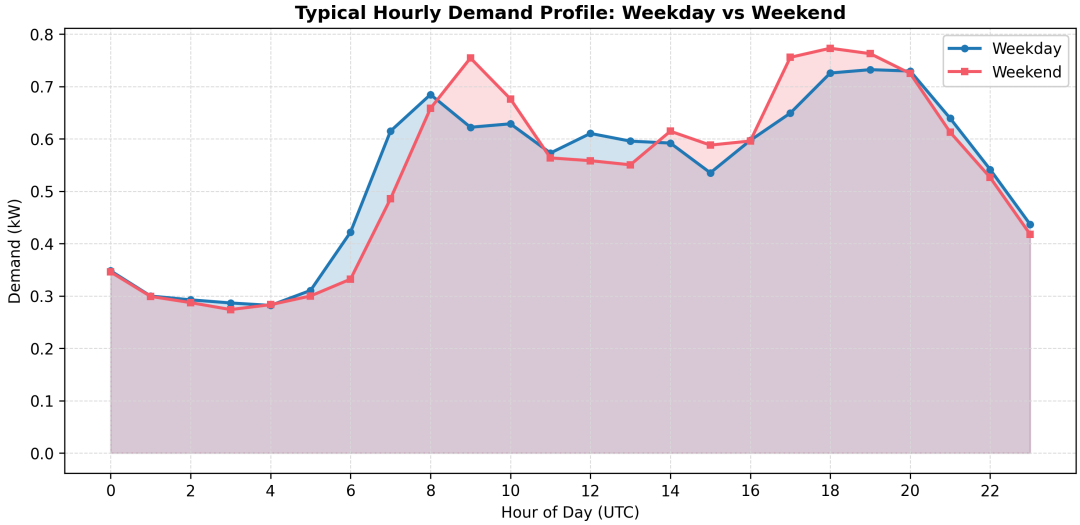

Figure 16: Typical hourly demand profiles: Weekday vs. Weekend. The weekday profile shows distinct morning and evening peaks, while weekends exhibit a flatter, delayed pattern.

These profiles directly inform the HEMS optimization strategy: the battery should discharge during evening peaks (17:00–21:00) when demand is highest and grid prices typically peak, while charging during midday hours when PV generation exceeds consumption.

7 Statistical Modeling and Time Series Analysis

7.1 Stationarizing the Demand Series

A fundamental prerequisite for ARIMA-family models is stationarity—the requirement that the statistical properties of the time series remain constant over time (Box et al., 2015). I assessed stationarity using two complementary tests:

- **Augmented Dickey-Fuller (ADF) Test:** Tests H_0: unit root exists (non-stationary). For the original demand series, the ADF statistic was -11.77 with p-value < 0.001, suggesting stationarity.
- **Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test:** Tests H_0: series is stationary. The KPSS statistic was 1.23 with p-value < 0.01, rejecting stationarity.

The conflicting results indicate deterministic trends or seasonality in the data. After applying first-order differencing ($d = 1$), both tests agree: the differenced series passes both ADF (p < 0.001) and KPSS (p > 0.05) criteria, confirming stationarity. This justifies using integrated

ARIMA models with $d = 1$.

7.2 ACF/PACF Analysis

The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the differenced series (Figure 17) reveal the correlation structure guiding model specification:

- **ACF:** Shows significant spikes at lags 24, 48, and 72, confirming strong daily (24-hour) seasonality in household demand patterns.
- **PACF:** Exhibits a sharp cutoff after lag 2–3, suggesting AR(2) or AR(3) non-seasonal components are appropriate.

These patterns motivate the use of Seasonal ARIMA (SARIMA) models with seasonal period $s = 24$ to capture the diurnal cycle.

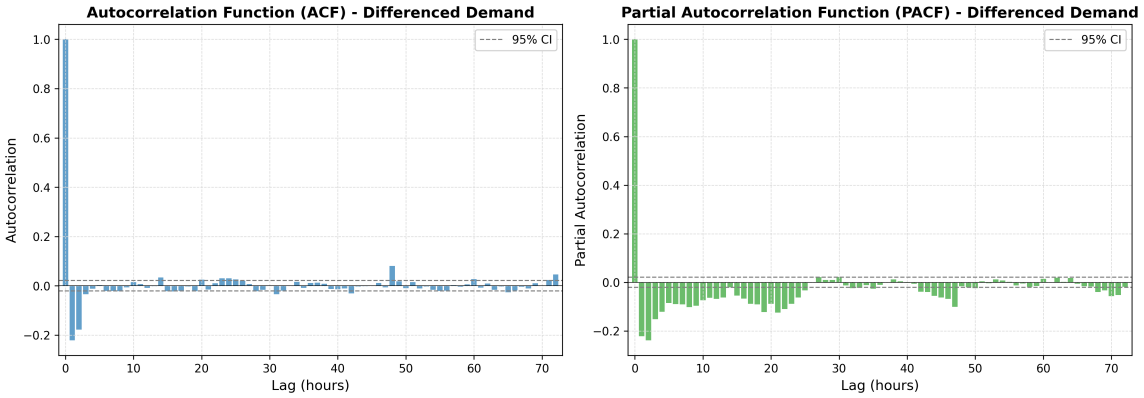

Figure 17: ACF and PACF plots of the differenced demand series. Significant spikes at lags 24, 48, and 72 confirm the 24-hour seasonal cycle. The PACF decay pattern suggests AR(2) components.

7.3 ARMA-Family Model Specification

Based on the ACF/PACF analysis, I evaluated three candidate models from the ARIMA family:

1. **ARIMA(2,1,2):** A non-seasonal model with second-order AR and MA terms, serving as a baseline.
2. **SARIMA(1,1,1)(1,1,1,24):** A full seasonal model with both seasonal AR and MA components.
3. **SARIMA(2,1,1)(0,1,1,24):** A simpler seasonal model with only the seasonal MA component ($Q = 1$).

7.4 Validation Methodology and nRMSE

I evaluated all models using the normalized Root Mean Squared Error (nRMSE), defined as:

$$\text{nRMSE} = \frac{\text{RMSE}}{\max(y) - \min(y)} \tag{7}$$

This scale-invariant metric allows comparison across different series magnitudes. I applied two validation strategies:

(1) Whole-Train Split: Training on all data up to a fixed cutoff, then evaluating on a single 24-hour window.

(2) Walk-Forward Validation: A rolling origin approach over 7 days where each day the model is refitted on all available history and evaluated on the next 24 hours.

Figure 18 shows the forecast overlay for the best model on the whole-train split.

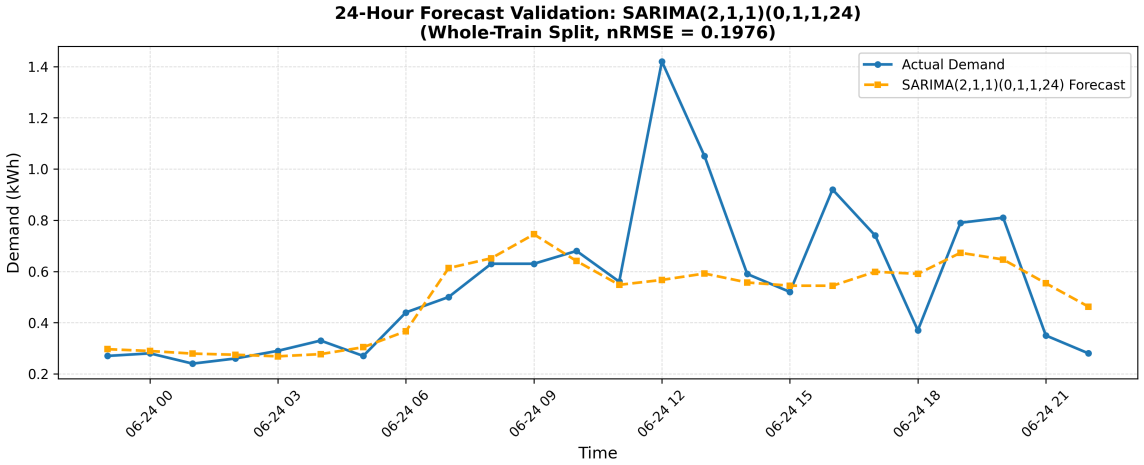

Figure 18: 24-hour forecast overlay: SARIMA(2,1,1)(0,1,1,24) vs actual demand. The model tracks the diurnal profile well but underestimates midday peaks.

7.5 Model Comparison and Selection

Figure 19 compares all models across both validation approaches. Key findings:

- Both SARIMA models significantly outperform the non-seasonal ARIMA, confirming the importance of explicitly modeling the 24-hour cycle.
- On the whole-train split, SARIMA(2,1,1)(0,1,1,24) achieves the lowest nRMSE (0.198).
- On walk-forward validation, SARIMA(1,1,1)(1,1,1,24) performs best (nRMSE = 0.275), indicating better generalization.
- I select **SARIMA(1,1,1)(1,1,1,24)** as the preferred model due to its superior walk-forward performance, which better reflects operational forecasting conditions.

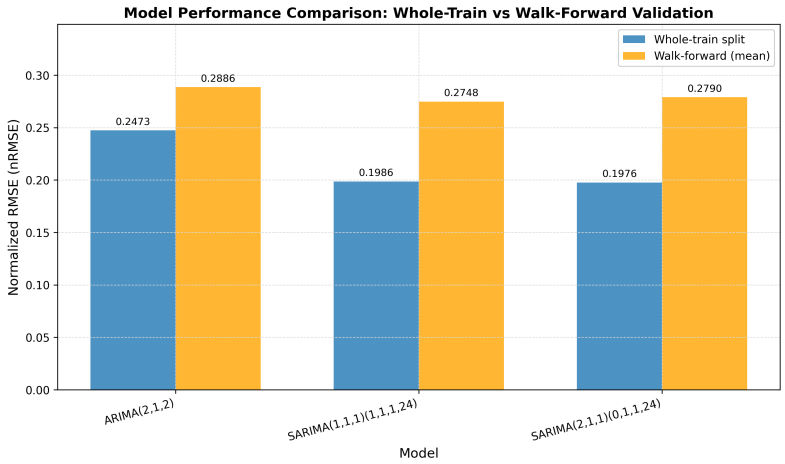

Figure 19: nRMSE comparison: SARIMA models outperform ARIMA(2,1,2). SARIMA(1,1,1)(1,1,1,24) achieves the best walk-forward performance.

8 Machine Learning Approaches

8.1 XGBoost Regression

To address the limitations of linear statistical models in capturing complex, non-linear interactions, I implemented an XGBoost (Extreme Gradient Boosting) regressor. XGBoost is a scalable implementation of gradient boosted decision trees, widely recognized for its state-of-the-art performance in structured data problems (Chen & Guestrin, 2016). Unlike ARIMA, which models temporal dependence through autoregressive terms, XGBoost requires explicit feature engineering to capture temporal structures.

8.2 Feature Engineering

I constructed a comprehensive feature matrix using the engineered features from Task 5:

- **Temporal Features:** Cyclical hour encodings (hour_sin, hour_cos), weekend indicator.
- **Weather Features:** Temperature, heating/cooling degree days, solar radiation components, cloud cover, wind speed, pressure.
- **Interaction Features:** Temperature-irradiance interaction to capture comfort-driven demand.
- **Price Signal:** Electricity price as an external regressor.

8.3 Hyperparameter Selection

Table 3 presents the selected hyperparameters and their rationale.

Table 3: XGBoost hyperparameters and selection rationale.

Parameter	Value	Rationale
n_estimators	600	Sufficient trees for convergence without excessive training time.
learning_rate	0.05	Moderate shrinkage balances speed and generalization.
max_depth	6	Allows capturing non-linear interactions without overfitting.
subsample	0.8	Row sampling adds regularization, reduces variance.
colsample_bytree	0.8	Feature sampling prevents over-reliance on single predictors.
reg_lambda	1.0	L2 regularization constrains model complexity.

8.4 Model Interpretation and Performance

The model was trained using a temporal train-test split (last 7 days held out) to prevent data leakage. Figure 20 displays the feature importance derived from the trained model. The cyclical hour encodings (hour_sin, hour_cos) are the most important predictors, confirming the strong diurnal pattern in demand. Temperature and heating degree days also rank highly, reflecting weather-driven consumption.

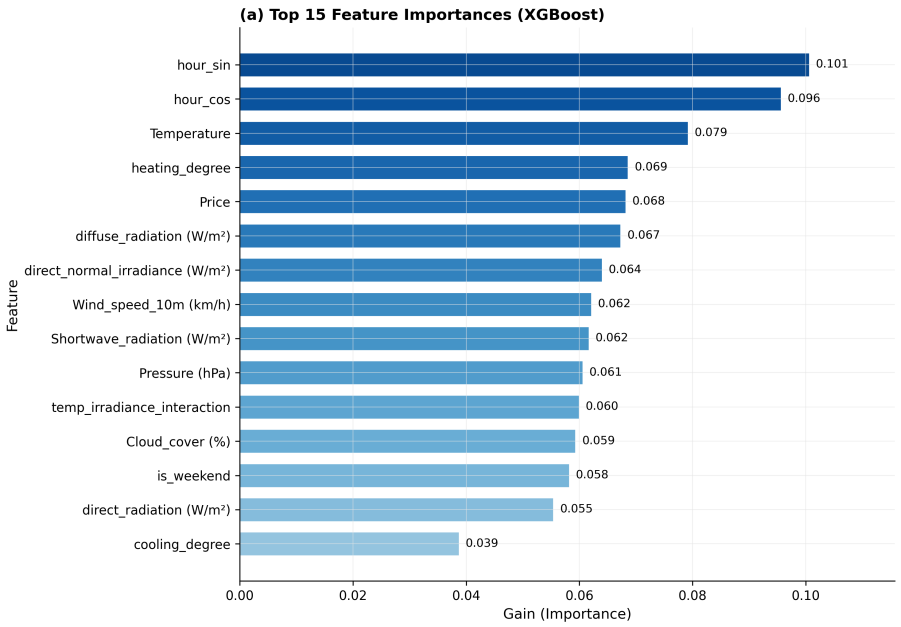

Figure 20: XGBoost feature importance. Cyclical hour features dominate, followed by temperature-related variables.

Figure 21 shows the forecast overlay for a representative day. The XGBoost model tracks the

demand profile reasonably well, though it struggles with sudden peak events that deviate from historical patterns.

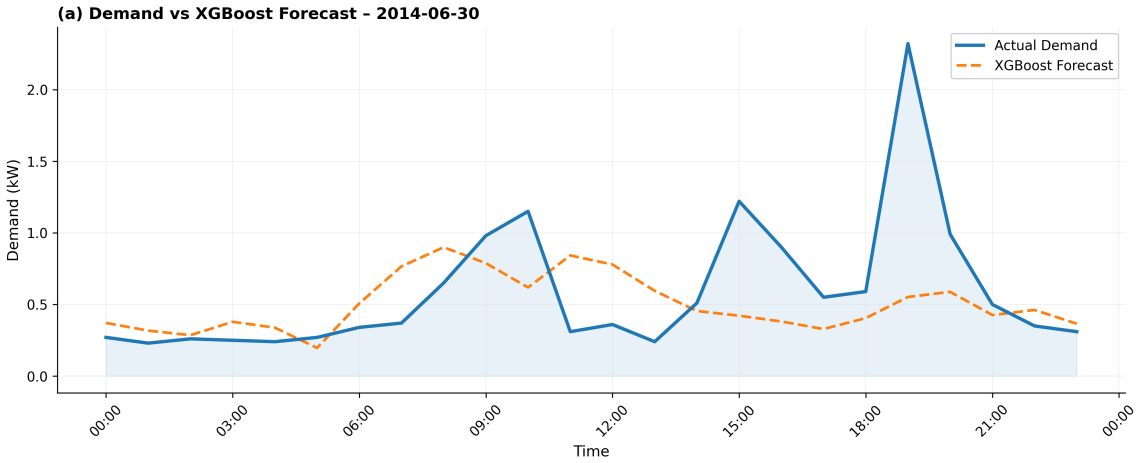

Figure 21: XGBoost forecast overlay for a representative day. The model captures the general diurnal pattern but underestimates peak demand.

8.5 Comparison with Statistical Models

Table 4 compares the XGBoost model against the best statistical model from Task 7.

Table 4: Performance comparison: XGBoost vs SARIMA.

Model	MAE	RMSE	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 the lowest nRMSE (0.166 vs 0.199), representing a 16.6% relative improvement over SARIMA.
- Despite higher absolute MAE, XGBoost's lower nRMSE indicates better performance relative to the demand range.
- The ML model benefits from exogenous features (weather, price) that SARIMA cannot directly incorporate in its univariate form.
- XGBoost's non-linear decision boundaries allow it to model complex interactions between features that linear models cannot capture.

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 Comparative Analysis

I compared the XGBoost model against two baseline models:

- **Naive:** Repeats the last observed value for all 24 forecast hours
- **Seasonal Naive:** Uses the same hour from the previous day

Figure 22 illustrates the forecast trajectories for a representative day. XGBoost captures the daily demand profile more effectively, particularly during morning and evening peak periods where the naive baselines systematically under- or overestimate demand.

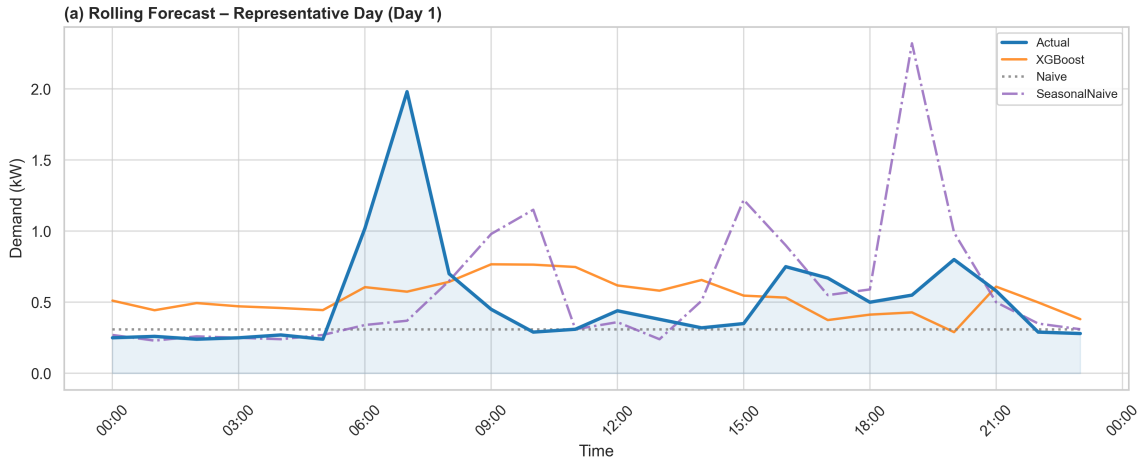

Figure 22: Forecast comparison for a representative day (Day 1). XGBoost tracks the actual demand pattern more closely than the baseline methods, especially during peak hours.

9.3 Seven-Day Forecast Performance

Figure 23 shows the complete 7-day forecast overlay for the best-performing model (XGBoost). The model successfully captures the general demand pattern across the evaluation period, though some extreme peaks (particularly on Day 7) present challenges.

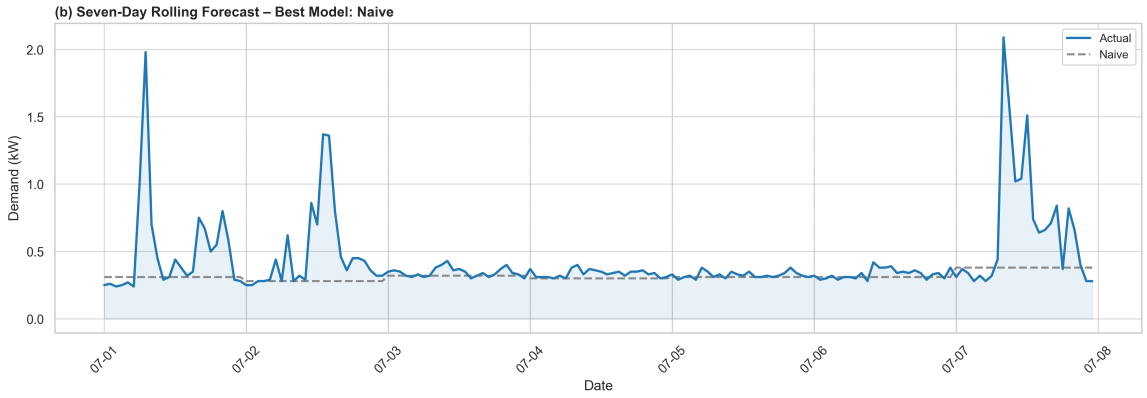

Figure 23: Seven-day rolling forecast overlay for XGBoost. The model tracks the general demand pattern effectively, with largest deviations occurring during unexpected demand spikes.

9.4 Aggregate Results

Table 5 summarizes the aggregate performance metrics across the 7-day evaluation period. XGBoost achieves the lowest normalized RMSE (nRMSE = 0.157), confirming its suitability as the primary forecasting model for the downstream battery optimization task.

Table 5: Aggregate forecast performance metrics (7-day rolling validation).

Model	MAE	RMSE	nRMSE
XGBoost	0.211	0.291	**0.157**
Naive	0.132	0.301	0.162
Seasonal Naive	0.185	0.384	0.207

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. 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 6 summarizes the performance comparison on the 7-day validation period.

Table 6: Performance comparison: AR-only vs. exogenous models.

Model	MAE	nRMSE	MAE Impr. (%)	nRMSE Impr. (%)
XGBoost (AR only)	0.175	0.140	–	–
XGBoost (with exog)	0.176	0.137	-0.63	**+1.86**
ARIMA (AR only)	0.243	0.168	–	–
ARIMAX (with exog)	0.235	0.165	+3.31	**+1.33**

The results show that exogenous variables provide modest but consistent improvements in nRMSE: 1.86% for XGBoost and 1.33% for ARIMAX. While the MAE improvement for XGBoost is negligible, ARIMAX shows a 3.31% MAE reduction. This suggests that exogenous features help reduce prediction variance, particularly during weather-driven demand anomalies.

10.3 Forecast Visualization

Figure 24 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.

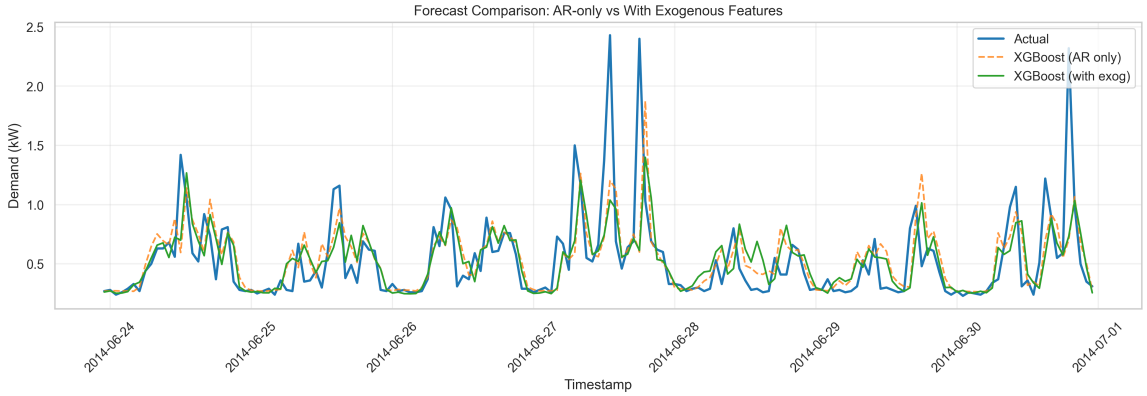

Figure 24: Forecast comparison: AR-only vs. exogenous models. XGBoost with exogenous features shows marginally improved tracking during variable demand periods.

10.4 Feature Importance Analysis

Figure 25 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.

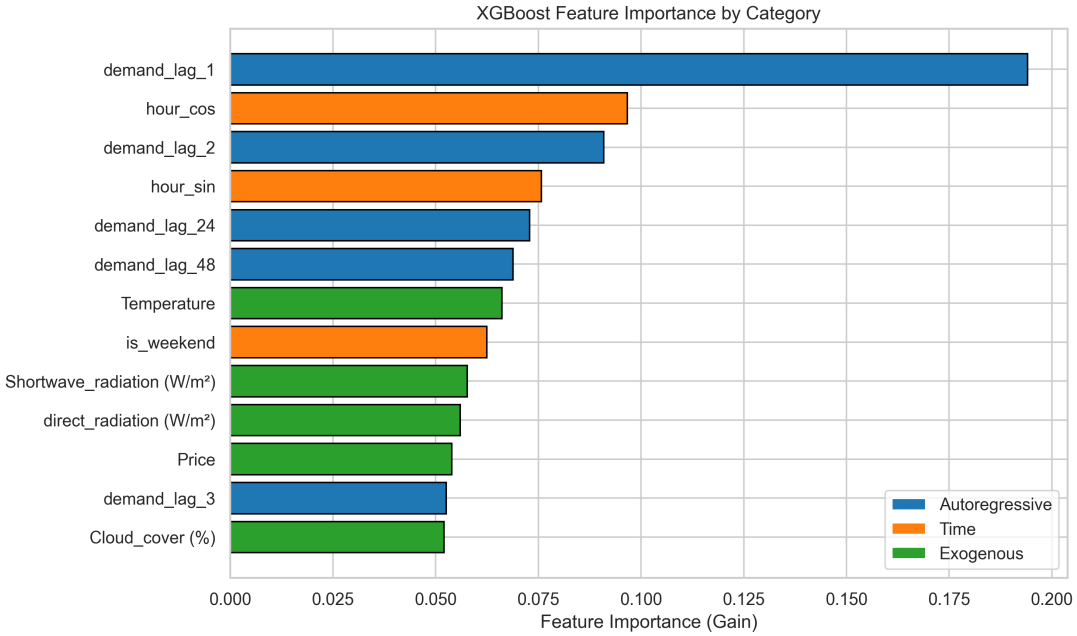

Figure 25: Feature importance by category. Autoregressive lags (blue) remain dominant, while exogenous weather features (green) provide supplementary predictive value.

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. Following the assignment specification, I formulated a Linear Programming (LP) problem to minimize the net electricity cost:

$$\min \sum_{t=1}^{n} (Gr_{c,t} - Gr_{P,t}) \tag{8}$$

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 7 summarizes the results.

Table 7: 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 26 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.

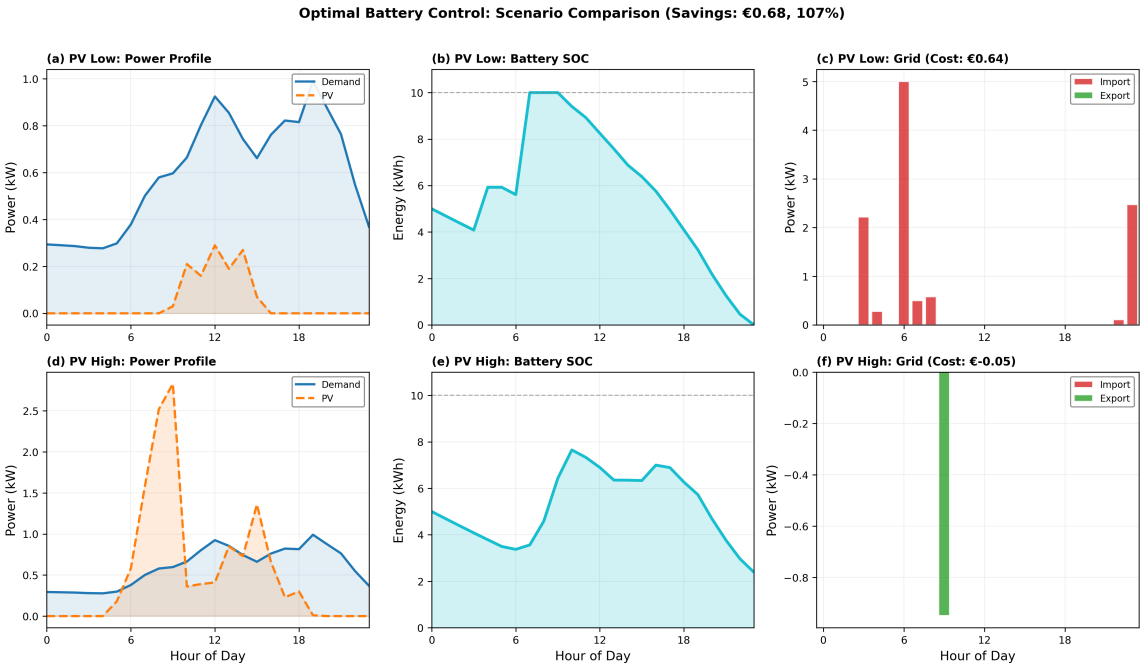

Figure 26: Optimal battery dispatch: PV_low (top) vs PV_high (bottom). Left: demand and PV profiles; center: battery SOC; right: grid exchange (positive = import, negative = export).

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

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

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

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

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

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

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

Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: Principles and techniques for data scientists* [Often cited as 2020 reprint]. O'Reilly Media.