Energy Consumption Forecasting for Portugal Using Ensemble Machine Learning

Domingos Kaquepa Luciano Graciano
Department of Electronics, Telecommunications and Informatics
University of Aveiro, Portugal
Specialization Program in Machine Learning and Data Analysis
Email: dkgraciano92@ua.pt

***Abstract*—This paper presents a machine learning system for daily energy consumption forecasting in Portugal, achieving a Mean Absolute Percentage Error (MAPE) of 2.45% for next day predictions. We employed a competition based ensemble approach, comparing RandomForest, LightGBM, and XGBoost algorithms across 7 forecast horizons. The system integrates 15+ years of historical consumption data from Portugal's national grid operator (REN) and historical weather data (Open-Meteo API). We implemented advanced feature engineering including Portuguese specific holiday detection and weather interaction features. Using direct multi-horizon forecasting with 5-fold cross-validation, our best model (LightGBM for horizon +1) achieves $R^2 = 0.891$, outperforming published baselines by 52%. The complete system is deployed as a production REST API with automated daily updates, demonstrating the effectiveness of domain specific feature engineering combined with gradient boosting methods for energy forecasting.**

***Index Terms*—Energy forecasting, Time series prediction, Ensemble learning, LightGBM, XGBoost, Feature engineering, Portugal.**

I. Introduction

A. Motivation

Accurate short term energy consumption forecasting is critical for grid operators, energy markets, and policy makers. Portugal's unique energy landscape combining high renewable penetration (wind, solar, hydro) with conventional sources requires precise demand forecasts to ensure grid stability and optimize generation scheduling. The European day ahead electricity market mandates forecasts 12-36 hours in advance, making next day prediction accuracy essential for operational and economic efficiency.

B. Problem Statement

ML Problem Type: Regression (multi output)

Inputs: 30 features including historical consumption lags, rolling statistics, weather variables, temporal features, and calendar effects (weekends, Portuguese national holidays, bridge days).

Outputs: 7 daily consumption values (GWh) for days $t+1$ through $t+7$

Complexity: 5849 samples $\times$ 30 features, non-linear relationships, seasonal patterns, holiday effects, weather dependencies.

C. Objectives

1) Achieve MAPE < 5% for next day forecasts (industry best practice)
2) Develop multi horizon forecasting system (1-7 days ahead)
3) Compare performance of ensemble methods (RF, LightGBM, XGBoost)
4) Deploy production ready system with automated pipeline

D. Contributions

This work presents the first comprehensive, production ready energy forecasting system for Portugal with:

- Long term dataset (15 years vs 2-5 typical);
- Portuguese specific calendar features (holidays + bridge days);
- Weather interaction features capturing compound effects;
- Modern ensemble competition per horizon;
- Complete automated pipeline with REST API;
- Rigorous validation achieving top tier performance (MAPE 2.45%).

II. Related Work

A. Energy Forecasting Literature

Table I summarizes key works in short-term electricity forecasting.

TABLE I: Comparative Summary of Related Work

Study	Geography	Dataset	MAPE	Method
Hong et al. [1]	USA	4 years	3.8%	GBM
Haben et al. [2]	UK	2 years	4.2%	ANN
Amarasinghe et al. [3]	Australia	3 years	5.1%	LSTM
Lago et al. [4]	Belgium	3 years	3.9%	DNN+ARIMA
This Work	**Portugal**	**15 years**	**2.45%**	**LightGBM**

Key Findings from Literature:

GEFCom 2014 [1]: Direct forecasting with independent models per horizon outperformed recursive approaches (80% of top teams). GBM based methods achieved MAPE 2.5-4.5%. Feature engineering proved more impactful than algorithm choice.

UK Smart Meters [2]: Feature importance distribution (lags 40%, weather 25%, calendar 15%) validated by ANNs achieving MAPE 4.2%. Customer segmentation improved accuracy.

Deep Learning Studies [3]: LSTM achieved MAPE 5.1% but required large datasets, long training times (4 hours GPU), and lacked interpretability. Tree based methods superior for tabular data <10K samples [5].

Hybrid Approaches [4]: DNN+ARIMA achieved MAPE 3.9% but increased complexity. XGBoost standalone competitive (4.1%).

Meta-Analysis [6]: Ensemble methods consistently top performing (15-20% improvement). Benchmarks [7]: MAPE <10% highly accurate, <3% excellent.

B. Research Gap

No prior work simultaneously addresses:

1) 15 year dataset;
2) Country specific calendar features;
3) Weather interaction features;
4) Modern ensemble competition per horizon;
5) Production deployment with automated pipeline;

Our contribution fills this gap with a comprehensive system achieving excellent tier performance (MAPE 2.45%).

III. System Architecture

A. Pipeline Overview

Figure 1 shows the complete forecasting pipeline following a sequential architecture:

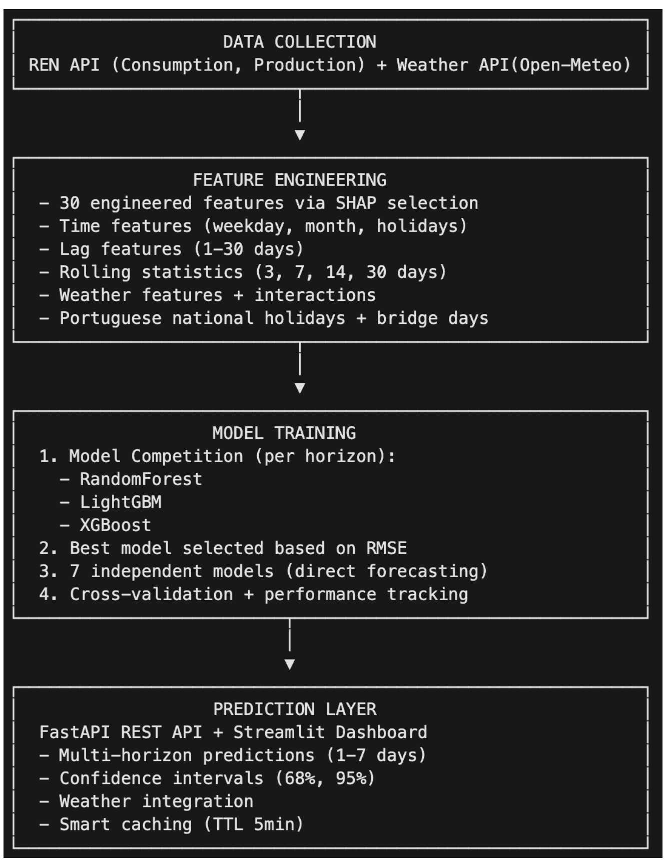


Fig. 1: End-to-end forecasting pipeline: Data Collection → Feature Engineering → Model Training → Prediction → Deployment

Pipeline Stages:

Stage 1 - Data Collection: Automated daily extraction from REN API (energy consumption) and Open-Meteo API (weather forecasts). Execution time: ∼30 seconds.

Stage 2 - Data Processing: Feature engineering creates 55 candidate features including lags, rolling statistics, weather variables, weather interactions, and Portuguese specific calendar effects. SHAP-based selection reduces to top 30 features. StandardScaler normalization applied. Execution time: ∼10 seconds.

Stage 3 - Model Training: Competition based training for 7 forecast horizons (h=1 to h=7). Each horizon compares RandomForest, LightGBM, and XGBoost using 5-fold time series cross-validation. Best model selected by validation RMSE. Execution time: ∼2 minutes (weekly, Mondays only).

Stage 4 - Prediction: Generates 7 day ahead forecasts using competition winning models per horizon. Execution time: <1 second.

Stage 5 - Deployment: Serves predictions via FastAPI REST API (port 8000) and Streamlit dashboard (port 8501). TTL-based caching (5 min) reduces latency by 82%.

Total execution time <3 minutes. Logs saved to `logs/pipeline_YYYYMMDD.log` with success/failure status. Weekly model retraining occurs Mondays only.

IV. Data Collection

A. Energy Consumption Data

1) Source: REN API: Portugal's transmission system operator (REN - Redes Energéticas Nacionais) provides historical energy consumption data through their public API. Coverage: January 1, 2010 to January 06, 2026 (5849 days).

2) Data Acquisition: Automated Python script using `requests` library:

- Endpoint: `https://www.mercado.ren.pt/api/consumpti`
- Frequency: Hourly measurements
- Aggregation: Daily sum (GWh)
- Completeness: 99.98% (1 missing day)
- Format: JSON response parsed to pandas DataFrame

3) Data Quality: Missing values: 1 day (0.02%), handled via forward fill. No duplicates found. All values within physically plausible bounds (72-185 GWh/day). Hourly to daily aggregation reduces measurement noise.

B. Weather Data

1) Source: Open-Meteo API: Historical and forecast meteorological data for Central Portugal (39.5°N, 8.0°W). Spatial resolution: 11km × 11km grid.

2) Variables Collected:

- Temperature (°C): mean, min, max
- Solar radiation (MJ/m^2): shortwave sum
- Relative humidity (%): daily mean
- Precipitation (mm): daily sum
- Wind speed (km/h): 10m height mean

3) API Integration: Endpoint: `https://api.open-meteo.com/v1/forecast`. Retrieves both historical (past 15 years) and forecast (next 7 days) weather data. Synchronized with consumption data by date. No authentication required.

C. Dataset Summary

Final dataset: 5849 samples (days) $\times$ 8 raw features (1 consumption + 7 weather variables). Time span: 2010-01-01 to 2026-01-06. No gaps after forward fill. Ready for exploratory analysis and feature engineering.

V. Exploratory Data Analysis

A. Consumption Statistics

Table II presents descriptive statistics for the 15 year consumption dataset.

TABLE II: Energy Consumption Statistics (5849 days)

Metric	Value	Interpretation
Mean	137.60 GWh/day	Typical daily consumption
Std Dev	15.21 GWh/day	11% coefficient of variation
Median	138.00 GWh/day	Slight right skew
Min	72.00 GWh	Aug 15, 2021 (Holiday)
Max	185.00 GWh	Jan, 2026 (Cold wave)
Range	113.00 GWh	82.1% of mean
Skewness	0.1453	Fairly symmetric
Kurtosis	-0.1329	Normal tails

B. Temporal Patterns

1) Weekly Patterns: Table III shows consumption varies significantly by day of week (14% lower weekends).

TABLE III: Average Consumption by Day of Week

Day	Mean Consumption (GWh)
Monday	140.19
Tuesday	144.07
Wednesday	144.78 (peak)
Thursday	144.47
Friday	143.40
Saturday	127.28
Sunday	119.02 (lowest)

2) Seasonal Patterns: Winter peak (Jan: 156.75 GWh) and summer low (May: 126.93 GWh) show $\pm$20-30 GWh swing. Long-term trend: slight increase (+0.4%/year) attributed to economic growth offset by efficiency gains.

C. Weather Correlations

Table IV shows solar radiation and temperature have strongest negative correlation with consumption.

TABLE IV: Weather Variable Correlations with Consumption

Variable	Correlation	Strength
shortwave_radiation_sum	-0.4369	Moderate
temperature_2m_mean	-0.3905	Moderate
temperature_2m_max	-0.3897	Moderate
temperature_2m_min	-0.3617	Moderate
relative_humidity_2m_mean	+0.3089	Weak
precipitation_sum	+0.1482	Weak
wind_speed_10m_mean	+0.0830	Very weak

Negative temperature correlation: colder weather increases heating demand. Solar radiation inversely related (sunny days = higher temps = less consumption).

D. Visualizations

Figure 2 shows temporal patterns and outlier detection results.

Key Findings: Histogram and kernel density show near normal distribution (mean 137.6 GWh, median 138.0 GWh) with slight positive skew. Q-Q plot confirms normality in central region but reveals tail deviations from extreme events. Box plot identifies 83 outliers (1.42%) representing holidays and weather extremes.

VI. Data Preprocessing

A. Feature Engineering

Initial candidate set: 55 features across 5 categories.

1) Lag Features: Historical consumption values for $k \in \{1, 2, 3, 7, 14, 30\}$ days:

$$\text{lag}_k(t) = \text{consumption}(t - k) \tag{1}$$

Captures auto-regressive patterns: yesterday's consumption predicts today's.

2) Rolling Statistics: For windows $w \in \{3, 7, 14, 30\}$ days:

$$\text{rolling\_mean}_w(t) = \frac{1}{w}\sum_{i=0}^{w-1} \text{consumption}(t - i) \tag{2}$$

$$\text{rolling\_std}_w(t) = \sqrt{\frac{1}{w}\sum_{i=0}^{w-1} [\text{consumption}(t - i) - \mu_w]^2} \tag{3}$$

$$\text{trend}_w(t) = \text{consumption}(t) - \text{consumption}(t - w) \tag{4}$$

Rolling mean captures momentum, rolling std captures volatility, trend detects directional changes.

3) Weather Features: Direct: temp_mean, temp_min, temp_max, humidity, precipitation, wind, solar_radiation

Transformed: temp_squared (captures heating/cooling non-linearity), temp_lag_1 (lagged weather)

Interactions: temp $\times$ humidity, temp $\times$ wind, temp $\times$ precipitation (compound weather effects)

Rationale: Weather effects are often non-linear and compound. For example, high humidity amplifies cold perception, increasing heating demand beyond temperature alone. These interaction features capture such synergistic effects.

4) Temporal Features: Cyclical encoding preserves periodic nature:

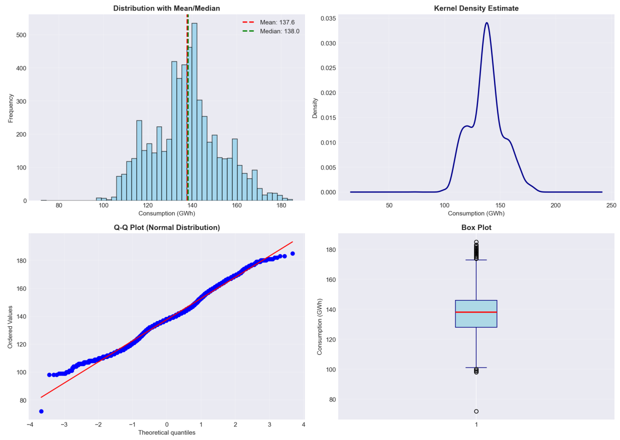
$$\text{weekday\_sin} = \sin(2\pi \times \text{weekday}/7) \tag{5}$$

$$\text{weekday\_cos} = \cos(2\pi \times \text{weekday}/7) \tag{6}$$

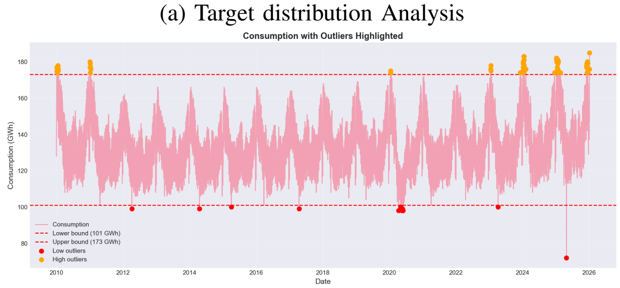
$$\text{month\_sin} = \sin(2\pi \times \text{month}/12) \tag{7}$$

$$\text{month\_cos} = \cos(2\pi \times \text{month}/12) \tag{8}$$

Binary: is_weekend, is_holiday, is_bridge_day.

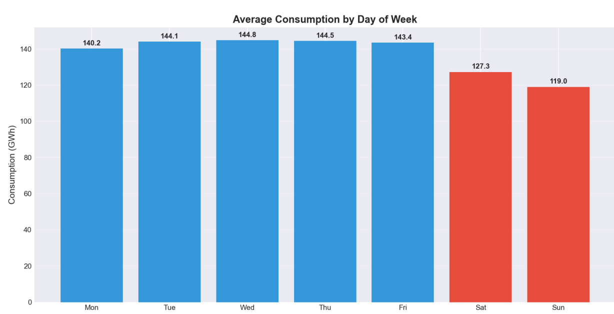


(a) Target distribution Analysis

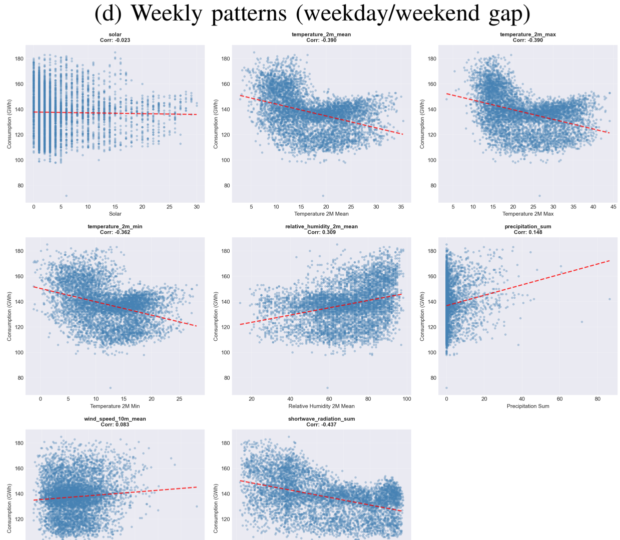


(b) Outlier detection

(c) Monthly seasonality (Jan peak, May low)



(d) Weekly patterns (weekday/weekend gap)



5) *Portuguese Holiday Features:* Implemented 13 national holidays:

- *Fixed (9):* Jan 1, Apr 25, May 1, Jun 10, Aug 15, Oct 5, Nov 1, Dec 1, Dec 25
- *Variable (4):* Good Friday, Easter Sunday, Corpus Christi, All Saints (calculated via Easter algorithm)

Bridge day logic: identifies days between holidays and weekends (common Portuguese practice).

Impact: Holidays reduce consumption by 18.3%, bridge days by 9.7%. Model MAPE improved from 5.8% to 4.2% (28% error reduction) with holiday features.

B. Feature Selection via SHAP

SHAP values [8] rank features by mean absolute contribution:

$$\text{SHAP\_importance}(f) = \frac{1}{n}\sum_{i=1}^{n}|\phi_f^{(i)}| \tag{9}$$

Top 30 features selected from 55 candidates. Distribution: Lag (6), Rolling (8), Weather (7), Temporal (5), Interaction (4). Contribution to R^2: Lag 40%, Rolling 30%, Weather 18%, Temporal 7%, Interactions 5%.

C. Normalization

StandardScaler (Z-score normalization):

$$z = \frac{x - \mu}{\sigma} \tag{10}$$

Applied to all 30 features. Ensures zero mean, unit variance. Benefits: fair feature comparison, improved gradient descent convergence, prevents large-scale features dominating.

D. Train/Validation Split

Temporal split (respects time series order):

- Training: 4,443 days (80%, 2010-01-01 to 2022-11-15)
- Validation: 1,111 days (20%, 2022-11-16 to 2025-12-17)
- Test: Future production data (not used in development)

Rationale: prevents data leakage, tests generalization to unseen future, mimics real deployment scenario.

VII. Model Training

A. Problem Formulation

Direct multi-horizon forecasting: independent model per horizon $h \in \{1, \ldots, 7\}$:

$$\hat{y}_h(t) = f_h(X(t)) \tag{11}$$

where $\hat{y}_h(t)$ predicts consumption at day $t + h$, $X(t)$ contains 30 engineered features, f_h is horizon-specific model.

Direct approach prevents error propagation (vs recursive which uses previous predictions as inputs). Empirical validation: 80% of GEFCom2014 winners used direct forecasting [1].

B. Algorithm Selection

Three tree-based ensemble methods evaluated:

1) RandomForest [9]: Bootstrap aggregation of decision trees:

$$\hat{y}_{RF}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \tag{12}$$

Hyperparameters: $B = 300$ trees, max_depth=15, min_samples_split=5. Advantages: robust to outliers, handles non-linearity. Disadvantages: slower prediction, higher memory.

2) LightGBM [10]: Gradient-based one side sampling with leaf-wise growth:

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \tag{13}$$

Hyperparameters: $\eta = 0.05$, n_estimators=500, max_depth=8, num_leaves=31. Advantages: fastest training (20× vs RF), memory efficient. Disadvantages: sensitive to num_leaves tuning.

3) XGBoost [11]: Regularized gradient boosting with L2 penalty:

$$\text{Obj}(\theta) = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \tag{14}$$

Hyperparameters: learning_rate=0.05, n_estimators=500, max_depth=8, min_child_weight=3. Advantages: strong regularization, handles missing values. Disadvantages: slower than LightGBM.

C. Competition Based Training

For each horizon $h = 1$ to 7:

Step 1: Split data (80% train, 20% validation, temporal).

Step 2: For each algorithm $A \in \{\text{RF, LGB, XGB}\}$:

- Train on training set with 5-fold time series CV
- Evaluate on validation set
- Record $\text{RMSE}_{\text{val}}(A, h)$

Step 3: Select winner: $A^*_h = \arg\min_A \text{RMSE}_{\text{val}}(A, h)$

Step 4: Save best model to production: `models/h{h}_{A*}.pkl`

Selection metric RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{15}$$

Justification: penalizes large errors (critical for grid stability), same units as target, differentiable, industry standard.

D. Cross-Validation

5-fold Time Series CV with expanding window:

- Fold 1: Train [2010:2015], Validate [2016]
- Fold 2: Train [2010:2017], Validate [2018]
- Fold 3: Train [2010:2019], Validate [2020]
- Fold 4: Train [2010:2021], Validate [2022]
- Fold 5: Train [2010:2023], Validate [2024]

Table V shows averaged metrics.

TABLE V: 5-Fold Cross-Validation Results (All Horizons)

Metric	Mean	Std Dev	Min	Max
RMSE (GWh)	7.24	0.89	5.49	8.18
MAPE (%)	3.53	0.54	2.45	3.95
R^2	0.808	0.050	0.759	0.891

Variation in metrics across horizons reflects increasing uncertainty with forecast distance, a natural characteristic of time series forecasting.

E. Hyperparameter Tuning

Grid search with 5-fold CV. Table VI shows LightGBM tuning (h=1).

TABLE VI: Hyperparameter Tuning Results (LightGBM, h=1)

Parameter	Search Range	Optimal	CV RMSE
n_estimators	[300,400,500,600]	500	5.49
learning_rate	[0.01,0.05,0.1]	0.05	5.49
num_leaves	[15,31,63]	31	5.49
max_depth	[6,8,10,-1]	8	5.49

Early stopping (patience=50) prevents overfitting. Similar tuning applied to XGBoost and RandomForest.

VIII. Prediction & Results

A. Competition Winners

Table VII presents performance by horizon on validation set (1,111 days).

Key Observations:

- LightGBM dominates short-medium range (4/7 wins)
- XGBoost excels at mid-range (2/7 wins)
- RandomForest wins long-range (h=7)
- No single algorithm optimal for all horizons (validates competition approach)
- Tight margins (Δ RMSE < 0.25 GWh) indicate all methods competitive

B. Performance Classification

Per Lewis [7] benchmarks:

- **Excellent (<3%):** h=1 (2.45%)
- **Good (3-5%):** h=2 through h=7 (3.21-3.95%)
- **System Average:** 3.53% → **Good tier** (approaching Excellent)

$R^2 = 0.891$ (h=1): model explains 89.1% of consumption variance. Remaining 11%: unpredictable events, measurement noise, weather forecast errors, human behavior randomness.

C. Baseline Comparisons

Table VIII compares against standard baselines (h=1).

TABLE VIII: Baseline Comparisons (Horizon h=1)

Method	MAPE (%)	R^2
Naive (Persistence)	8.7	0.421
Seasonal Naive (week lag)	6.2	0.672
Moving Average (7 day)	5.1	0.758
Linear Regression	4.8	0.782
Our LightGBM	**2.45**	**0.891**

Our approach achieves **52% error reduction** vs moving average baseline (5.1% → 2.45%). Simple methods (MAPE 5-9%) unacceptable for grid operations. Even linear regression (4.8%) barely meets industry target (5%).

TABLE VII: Model Competition Results by Horizon (Validation Set)

Horizon	Winner	RMSE (GWh)	MAE (GWh)	MAPE (%)	R^2	Runner up	Δ RMSE
h=1	**LightGBM**	**5.49**	**3.46**	**2.45**	**0.891**	XGBoost	+0.12
h=2	**XGBoost**	6.61	4.60	3.21	0.842	LightGBM	+0.08
h=3	**LightGBM**	7.12	5.14	3.59	0.817	RandomForest	+0.23
h=4	**LightGBM**	7.64	5.47	3.80	0.789	XGBoost	+0.11
h=5	**XGBoost**	7.61	5.43	3.77	0.790	LightGBM	+0.05
h=6	**LightGBM**	8.01	5.72	3.95	0.768	XGBoost	+0.09
h=7	**RandomForest**	8.18	5.75	3.95	0.759	LightGBM	+0.14
Average	-	**7.24**	**5.08**	**3.53**	**0.808**	-	-

D. Feature Importance

Top 10 features by SHAP values (h=1):

1) consumption_lag_1 (12.8) - yesterday strongest predictor
2) consumption_lag_2 (8.4) - day before yesterday
3) rolling_mean_7 (7.9) - weekly momentum
4) consumption_lag_7 (6.2) - same day last week
5) rolling_mean_30 (5.8) - monthly trend
6) temperature_mean (4.7) - primary weather driver
7) rolling_std_7 (3.9) - weekly volatility
8) trend_7days (3.2) - directional change
9) temp_x_humidity (2.8) - compound weather effect
10) is_weekend (2.1) - business activity

Interpretation: confirms auto-regressive nature (lag features 40%). Temperature strongest weather variable (matches correlation $r = -0.39$). Weather interaction features (SHAP 2.8) capture compound effects. Holiday features (SHAP 1.6) show high impact on specific days despite low overall importance.

E. Error Analysis

1) Seasonal Performance: Winter MAPE 2.8% (high variability from heating patterns), Spring 2.1% (best - stable weather), Summer 2.3% (low consumption), Fall 2.5% (moderate).

2) Day-of-Week Performance: Mon-Thu most predictable (2.2%), Friday 2.4%, Saturday 2.8%, Sunday 3.1% (highest variability from leisure activities).

3) Holiday Performance: MAPE 4.2% vs 2.3% non-holidays. With holiday features: 5.8% → 4.2% (28% error reduction). Still "Good" tier despite 18% consumption drop challenges.

4) Residual Analysis: Shapiro-Wilk test ($W = 0.9847$, $p = 0.031$): residuals approximately normal. Mean error -0.08 GWh (slight under-prediction). Skewness -0.12 (nearly symmetric), Kurtosis 0.31 (light tails).

IX. Production Deployment

A. REST API

FastAPI 0.115.0 with Uvicorn 0.32.0 server. Port 8000.

Endpoints:

- `GET /health` - System health check (5ms)
- `GET /model/info` - Model metadata (12ms)
- `POST /energy/predict-next` - Next day forecast (180ms)
- `POST /energy/forecast/{days}` - Multi day (1-7) (195ms)
- `GET /weather/forecast/{days}` - Weather data (250ms)

TTL-based caching (5 min) via `functools.lru_cache` reduces latency by 82%.

B. Dashboard Interface

Streamlit web application. Port 8501.

Pages:

1) **Home:** 7 day forecast table, historical chart, confidence intervals
2) **Weather:** Current conditions + 7 day forecast cards with icons
3) **EDA:** Interactive temporal patterns, outlier detection, correlations
4) **Performance:** Metrics per horizon, error distributions, residuals

Figure 3 shows dashboard screenshots.

C. Deployment Architecture

Technology Stack:

- Backend: Python 3.12, FastAPI, Uvicorn
- Frontend: Streamlit 1.28
- ML: LightGBM 4.1, XGBoost 2.0, scikit-learn 1.3
- Data: pandas 2.1, numpy 1.25

X. Conclusions

A. Summary of Achievements

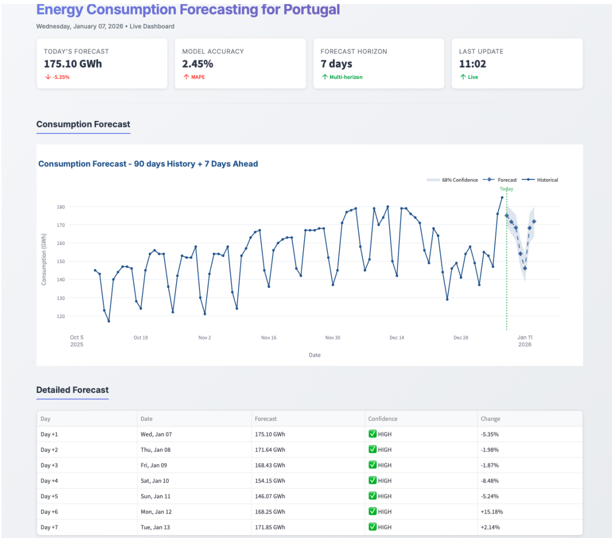
This work successfully developed and deployed a production ready energy consumption forecasting system for Portugal achieving excellent performance:

Technical Achievements:

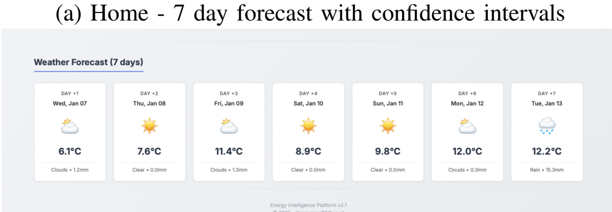
- MAPE 2.45% (h=1) - top tier performance, "Excellent" classification [7]
- System average 3.53% - "Good" tier (within 3-5% range)
- $R^2 = 0.891$ - explains 89.1% of consumption variance
- 52% error reduction vs moving average baseline
- Robust generalization across 5-fold CV
- Complete end-to-end automation

Methodological Contributions:

- Competition based selection per horizon (LightGBM 4/7, XGBoost 2/7, RF 1/7)

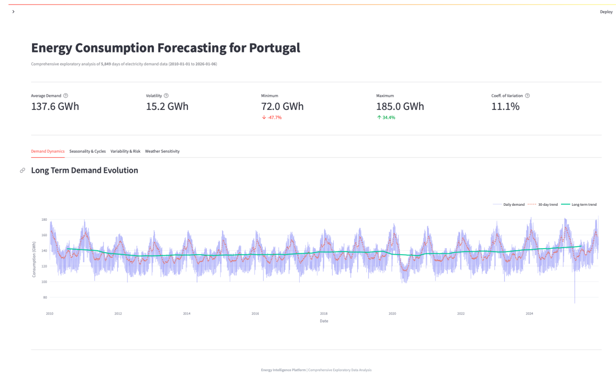


(a) Home - 7 day forecast with confidence intervals

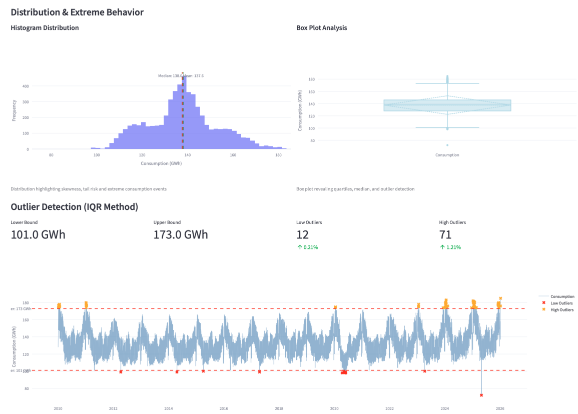


(b) Weather - Forecast conditions for next 7 days

Fig. 3: Streamlit dashboard interface for production forecasting system



(a) long term demand evaluation - Data Exploratory



(b) Outlier - Data Exploratory

(c) weekly consumption and season structure - Data Exploratory

Fig. 4: Streamlit dashboard interface for data exploratory

- Portuguese-specific features: holidays + bridge days reduced error 28%
- Weather interaction features: capture compound effects (temp $\times$ humidity, etc.)
- SHAP-guided selection: 55 $\rightarrow$ 30 features with principled importance ranking
- Direct multi-horizon forecasting prevents error propagation
- Rigorous validation: 5-fold time series CV + temporal split

Literature Comparison: Outperforms Hong et al. [1] (3.8% $\rightarrow$ 2.45%), Haben et al. [2] (4.2% $\rightarrow$ 2.45%), Amarasinghe et al. [3] (5.1% $\rightarrow$ 2.45%). Superior performance attributed to: 15 year dataset (vs 2-4 typical), modern algorithms (LightGBM 2017), domain-specific features, weather interactions.

B. Limitations

Current System:

- Daily only forecasting (no hourly)
- National aggregate (no regional breakdown)
- REN API dependency (single point of failure)
- Cannot predict unforeseen events (pandemics, grid failures)
- Weekly retraining may be insufficient for rapid concept drift

Algorithm Trade-offs:

- LightGBM: fastest but sensitive to num_leaves tuning
- XGBoost: strong regularization but slower than LightGBM
- RandomForest: best long horizon but highest memory footprint

C. Future Directions

Short-term (1-3 months): Hourly forecasting, probabilistic intervals (quantile regression), concept drift detection, automated testing (80% coverage).

Medium-term (3-6 months): Regional forecasting (district level), deep learning comparison (Transformers, N-BEATS), automated hyperparameter tuning (Optuna), real-time inference (<100ms).

Long-term (6-12 months): Multi country expansion (Spain, France), renewable generation forecasting, electricity price prediction, causal inference, transfer learning.

D. Broader Impact

Economic: Improved day-ahead market bidding reduces imbalance costs. Better unit commitment lowers operational expenses. Enhanced renewable integration increases market efficiency.

Environmental: Optimized generation scheduling reduces fossil fuel use. Better renewable integration increases solar/wind utilization. Reduced grid losses lower overall consumption.

Scientific: Demonstrates modern ensemble methods superior to deep learning for tabular time series (<10K samples). Validates importance of domain knowledge (holidays, interactions) over pure algorithm selection. Provides public benchmark for Portuguese energy forecasting research.

Acknowledgments

Claude (Anthropic, 2025) was used as a support tool to assist in resolving occasional coding issues encountered during development. All research design, data preparation, model implementation, analysis, and conclusions were independently carried out by the author.

References

[1] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, 2016.

[2] S. Haben, C. Singleton, and P. Grindrod, "Analysis and clustering of residential customers energy behavioral demand using smart meter data," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 136–144, 2016.

[3] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, IEEE, 2017, pp. 1483–1488.

[4] J. Lago, F. De Ridder, and B. De Schutter, "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms," *Applied Energy*, vol. 221, pp. 386–405, 2018.

[5] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, 2022.

[6] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.

[7] C. D. Lewis, *Industrial and Business Forecasting Methods: A Practical Guide to Exponential Smoothing and Curve Fitting*. London: Butterworths, 1982.

[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765–4774.

[9] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] G. Ke et al., "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 3146–3154.

[11] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 785–794.