Electricity Demand Forecasting

Fiona Chow
Center for Data Science
New York University
fc1132@nyu.edu

Patrick Su
Center for Data Science
New York University
zs1512@nyu.edu

Abstract

Electricity demand forecasting is crucial for efficient energy planning, grid stability, and the integration of renewable energy sources. This project forecasts daily average electricity demand in the London region using ARIMA-based models and Gaussian Processes. Empirical results show that ARIMA models effectively forecast patterns but face increasing uncertainty over longer horizons, while Gaussian Processes offer flexibility to model nonlinear trends and multiple seasonalities with calibrated uncertainty estimates. Exogenous variables, such as weather data, are incorporated to improve predictive accuracy, demonstrating the advantages of combining multiple kernels and external factors.

1 Introduction

In the context of increasing energy consumption and the integration of renewable sources([1]) , accurate forecasting electricity demand allow utilities to manage load balancing, schedule power generation, and develop dynamic pricing strategies. The goal of this project is to evaluate ARIMA-based models and Gaussian Processes on an electricity demand dataset to see how they differ in forecasting.

1.1 Data

The dataset we are using is an existing electricity demand and weather dataset from Hugging Face [1] which harmonizes multiple open smart meter datasets. Specifically, we focus on the London region, which contains 30-minute frequency data on electricity consumption for residential buildings. The dataset spans from November 23, 2011, to February 28, 2014, covering a total of 2.27 years.

Dataset Structure The dataset comprises three components:

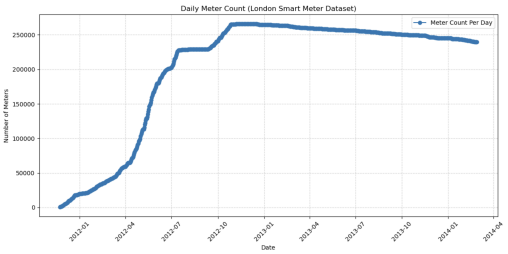
- **Electricity Consumption Data:** Includes unique meter IDs, timestamps, and consumption in kWh.
- **Metadata:** Describes each meter's location, building class, data frequency, and geohash.
- **Weather Data:** Provides hourly observations such as temperature, humidity, precipitation, wind speed, and radiation for each location.

1.2 Design Choices

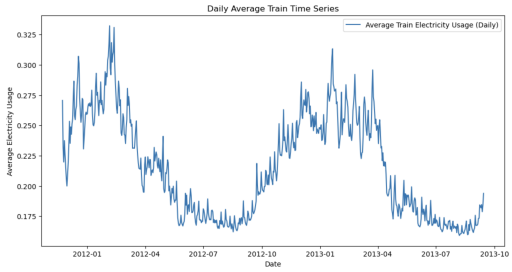
Average (Per Meter) Electricity Demand We chose to forecast the daily average electricity demand (per meter) rather than the total electricity demand. This is to address the increasing daily meter count over time (see Figure 1a). Modeling average meter consumption normalizes the data

[1] https://huggingface.co/datasets/EDS-lab/electricity-demand

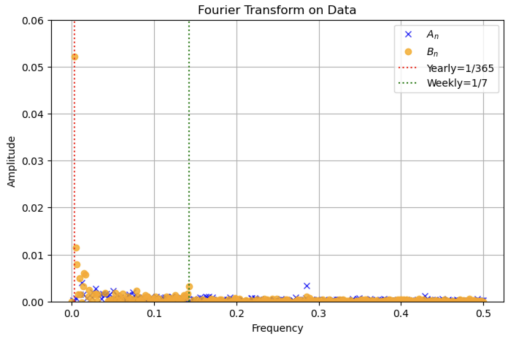
by accounting for the varying number of meters, ensuring that the analysis focuses on per-meter consumption trends rather than being skewed by changes in the total meter count.



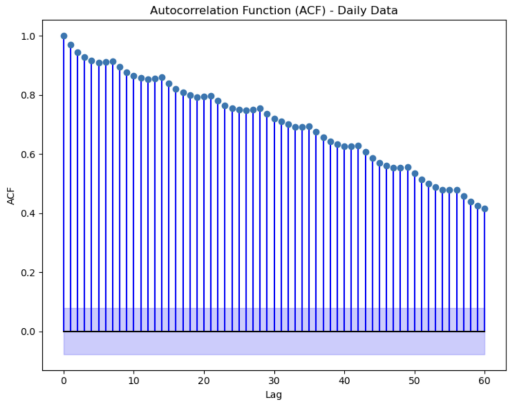
(a) Meter counts over time



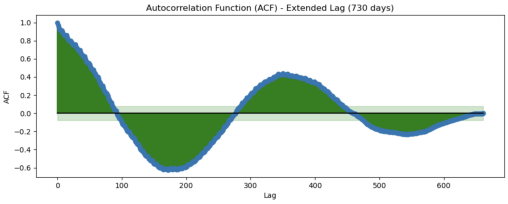
(b) Daily average time series of data



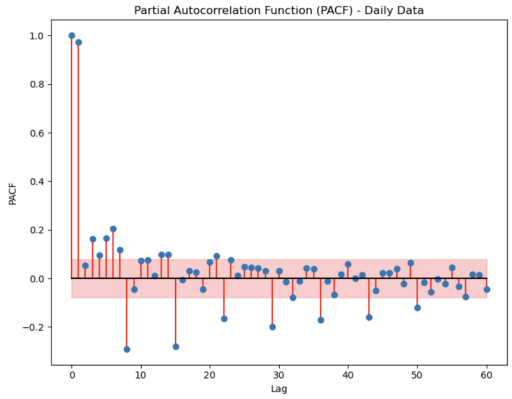
(c) Fourier Transform on Data



(d) ACF



(e) Extended lag ACF



(f) PACF

Daily Level Forecasting We chose to forecast daily electricity demand to simplify the problem by removing intra-day variations, enabling clearer interpretation of weekly and yearly trends, and allowing quicker model fit times.

The plot in Figure 1b illustrates the daily average time series of our data.

1.3 Data Exploration

Seasonality Our data exhibits two primary seasonal patterns: weekly and yearly. This is evident from Figure 1c, where the Fourier Transform highlights peaks at the yearly and weekly frequencies. Additionally, Figure 1d reveals a weekly seasonality, as seen in the recurring spikes at multiples of 7 lags in the Autocorrelation Function (ACF). Similarly, Figure 1e confirms yearly seasonal pattern with broader peaks at lags corresponding to annual cycles. The ACF effectively captures seasonality by quantifying correlations with past values, making periodic patterns clearly visible.

Stationarity To determine whether our data was stationary, we performed an Augmented Dickey-Fuller (ADF) test. The ADF test is a statistical test used to determine whether a time series is stationary by checking for the presence of a unit root, with a low p-value indicating stationarity. Initially, the ADF test yielded a p-value of 0.409, indicating that the null hypothesis of a unit root could not be rejected and the data was non-stationary. To address this, we applied first-order differencing to the data, which removed the trend and made the series stationary. After differencing, the ADF test returned a p-value of 4.47×10^{-9}, strongly rejecting the null hypothesis and confirming that the transformed data is now stationary

2 Methods

We started with a simple ARMA model and progressively introduced additional complexity and flexibility to our modeling process.

2.1 ARMA

Based on the visual inspection of the Partial Autocorrelation Function (PACF) and ACF, we selected ARMA(1,0), indicating $p = 1$ autoregressive term and $q = 0$ moving average term.

AR order As shown in Figure 1f, the PACF drops within the error bands by lag 2, indicating that only the first lag has a significant direct influence on the current value. This pattern aligns with an AR(1) process. The PACF isolates direct lag effects, making it effective for identifying AR processes where noise propagates over time. However, in this simpler case of AR(1), such noise propagation is not observed.

MA order In contrast, MA processes are noiseless beyond the specified lag, leading to a sharp cut-off in the ACF. However, in Figure 1d, the absence of a sharp cut-off suggests that either there is no MA component or its influence is masked by the AR process. As a result, we modeled the data with an MA(0) component.

2.2 ARIMA

Next, we ran the AutoARIMA package to identify the optimal ARIMA parameters and compare results to our visually inspected ARMA (see results in Appendix A). AutoARIMA is an automated algorithm that selects the best ARIMA model by optimizing its parameters (p, d, q) and seasonal components (if applicable) based on criteria like Akaike Information Criterion (AIC).

The package selected ARIMA(2,1,1), indicating $p = 2$ autoregressive terms, $d = 1$ differencing step, and $q = 1$ moving average term.

2.3 SARIMA

To incorporate the seasonality observed during data exploration described in Section 1.3, we applied SARIMA, which is represented by the following equation:

$$\Phi_p(B^s)\phi_p(B)(1-B)^d(1-B^s)^D y_t = \Theta_q(B^s)\theta_q(B)\varepsilon_t$$

SARIMA extends ARIMA by incorporating seasonal terms to model periodic patterns in data. Specifically:

- B is the backshift operator ($By_t = y_{t-1}$), and B^s applies seasonal lags (s periods).
- d and D represent non-seasonal and seasonal differencing orders, ensuring stationarity by removing trends and seasonal cycles.
- p, q, P, Q are the orders of non-seasonal (p, q) and seasonal (P, Q) AR/MA polynomials, capturing regular and periodic patterns.
- The constants Φ_p and Θ_q are the coefficients for the seasonal AR and MA components. Similarly, ϕ_p and θ_q are the coefficients for the non-seasonal AR and MA components. The residual ε_t captures the unexplained variation at time t.

In order to determine parameters, we ran the AutoARIMA package with the `seasonal` flag set to `True`, testing separately [2] for 7-day and 365-day seasonal periods.

Weekly seasonality The package selected SARIMA(1,1,1)(2,0,2,7) indicating non-seasonal terms $p = 1, d = 1, q = 1$, seasonal terms $P = 2, D = 0, Q = 2$, and a seasonal period of $s = 7$.

Yearly seasonality We ran a similar set up for yearly but the kernel crashed several times and in the interest of time, moved on to the next model to try to capture this seasonality in a different way.

2.4 SARIMAX

To incorporate yearly seasonality, we implemented SARIMAX. SARIMAX extends SARIMA by incorporating exogenous variables and is represented by the following equation:

$$\Phi_p(B^s)\phi_p(B)(1-B)^d(1-B^s)^D y_t = \Theta_q(B^s)\theta_q(B)\varepsilon_t + \beta X_t$$

where X_t are external covariates at time t, and β is the coefficient that captures the effect of these variables on the target variable y_t.

We incorporated weather data as the exogenous variable. The backbone parameters remained consistent with those of the SARIMA model discussed earlier.

2.4.1 Fine-Tuned Model Optimization

Weather Variables We fined tuned our SARIMAX model to find the best combination of weather variables. We plotted the correlation matrix (see Table 4 in Appendix C) of electricity demand and weather variables and picked the top five correlated weather variables (see Table 2 in Appendix C) and did a grid search of varying combinations of them to find the best set of weather variables based on MAPE performance (see Table 3 and 4 in Appendix C).

Other Parameters Simiarly, we searched over a range of parameters (see Listing 1 in Appendix C) to find the best p, q, P, Q, D while using the best set of weather variables we had just found. We selected our model based on best MAPE performance. This ended up to be SARIMA(1,1,2)(0,1,2,7).

Confidence Interval Adjustment Finally, we reduced our confidence intervals from 95% confidence (2 standard deviation) to 68% (1 standard deviation) to allow for narrower bands.

2.5 Gaussian Processes

To capture the underlying patterns, we focused on two key kernels: the squared exponential kernel (or RBF), which models smooth variations in demand,

$$k(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

and a periodic exponential kernel, which captures the seasonality in consumption.

$$k(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{2\ell^2}\right)$$

Using the fact that the sum of two kernels is also a kernel, we are able to test different combinations of kernels and a detailed exploration can be found in the results section.

Parameters For example, a weekly periodic exponential kernel will set the period parameter to $p = 7$. The lengthscale ℓ parameter is the main parameter we experiment during the fine-tuning process. While we also consider tuning the variance, the results used in this paper all assume variance $\sigma^2 = 1.0$ Since we are using the GPy package for development, we also compare the results from model.optimize and model.optimize_restarts (using restart to reduce the likelihood of the optimizer getting stuck in a local minimum) to see if there is any tradeoff between model fitting and forecasting results (we set $random\_state = 42$ for reproducibility)

[2] SARIMA cannot effectively capture multiple seasonalities simultaneously ([2])

Other kernels Other kernels, such as white noise kernels and yearly periodic kernels, are also considered. However, under the current assumption, no major effects are observed, so they are not included in this report but can be easily reproduced follow the notebooks in the code repository.

Weather Variables To build a weather RBF kernel [3], we consider two features ('temperature_2m', 'soil_temperature_7_to_28cm', two features that have very high correlation score) from the weather dataset for simplicity (due to time constraint, we could not test out all combinations)

3 Results

For evaluation, we split the data into train and test sets using an 80/20 chronological split. The models were assessed using model fit and forecast performance metrics to evaluate training performance and predictive accuracy on unseen data, ensuring effective generalization.

Model Fit Metrics Model fit metrics include log-likelihood, which measures how well the model explains the training data—a higher log-likelihood indicates a better fit. Additionally, we used AIC, which balances model fit and complexity by penalizing the number of parameters. Lower AIC values signify a better model when comparing alternatives.

Forecast Performance Metrics Forecast performance metrics include Mean Absolute Percentage Error (MAPE), which calculates the average percentage deviation between predictions and actual values, and Root Mean Squared Error (RMSE), which quantifies the magnitude of forecast errors. Lower values for both metrics indicate better predictive accuracy on unseen test data.

3.1 Results

Results are summarized in Table 1 and plots in Figure 2.

Table 1: Model Comparison: Forecast Performance and Model Fit

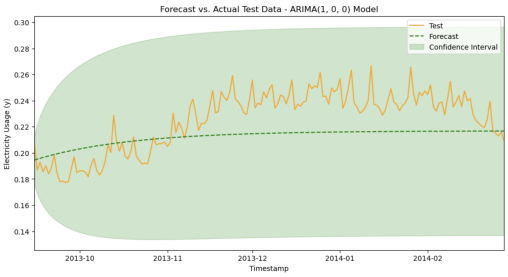
Model	Forecast Performance (Test Data)		Model Fit (Training Data)	
	MAPE(%)	RMSE(kWh)	Log-Likelihood	AIC
ARIMA(1, 0, 0)	8.420	0.0224	2119	-4232
Auto ARIMA(2, 1, 1)	16.41	0.0443	2136	-4263
Auto SARIMA(1, 1, 1)(2, 0, 2, 7)	10.31	0.0285	2227	-4509
SARIMA(1, 1, 1)(2, 0, 2, 7) + Weather	4.870	0.0134	**2311**	**-4605**
Final Tuned SARIMA(1, 1, 2)(0, 1, 2, 7) + Weather	**2.900**	0.0083	2304	-4588
GP (Weekly Periodic Kernel Only)	8.42	0.0005	-316	647
GP (Demand RBF Kernel Only)	9.86	0.0006	-94	193
GP (Weather RBF Kernel Only)	6.49	0.0003	-195	395
GP (Weekly Periodic + Demand RBF)	13.71	0.0012	21	-31
GP (Weekly Periodic + Demand RBF + Weather)	5.27	0.0002	206	-397
GP (Weekly Periodic + Demand RBF + Weather without restart)	4.7	0.0002	42	-69

3.2 Discussion

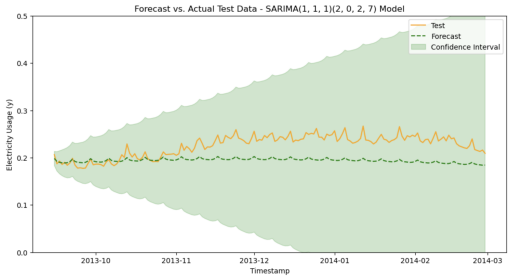
ARMA We compared the theoretical ACF of ARMA(1,0) (see Figure 3 in Appendix B) with the estimated ACF (Figure 1d) to assess the model's ability to capture the data's autocorrelation structure. The similarity between the two confirms a good model fit. Additionally, MAPE is 8.42%, supporting ARMA(1,0) as a strong starting point for modeling London's daily average electricity demand. However, Figure 2a shows that the forecast neither tracks the weekly nor yearly seasonality in the test set.

SARIMA SARIMA(1,1,1)(2,0,2,7) captured the weekly trends (see Figure 2b) in the forecast that ARMA(1,0) did not. The model fit on train data are also better along with the forecast performance (Mape is 10.31%) on test data.

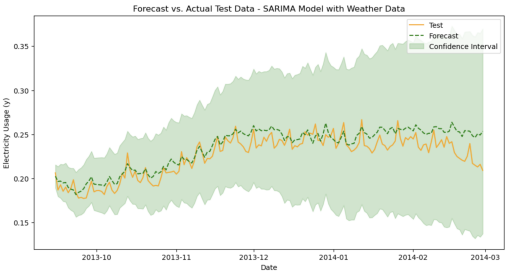
SARIMAX SARIMA(1,1,1)(2,0,2,7) model that incorporated weather data as the exogenous variable performed very well with great model fit scores on training data and MAPE of 4.87% on test data. The model not only captured the weekly trends but also captured the yearly trends (see Figure 2c.



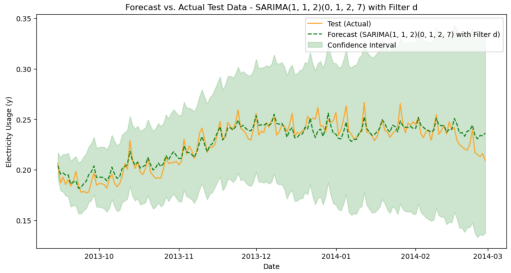
(a) ARMA(1,0)



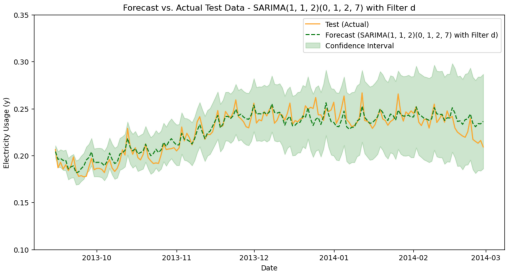
(b) SARIMA(1,1,1)(2,0,2,7)



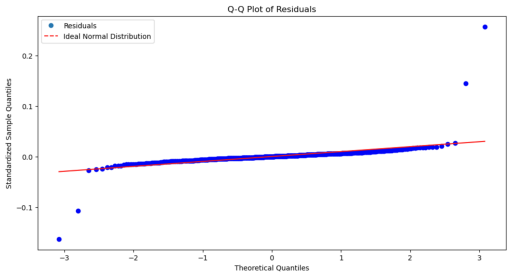
(c) SARIMA(1,1,1)(2,0,2,7) + Weather



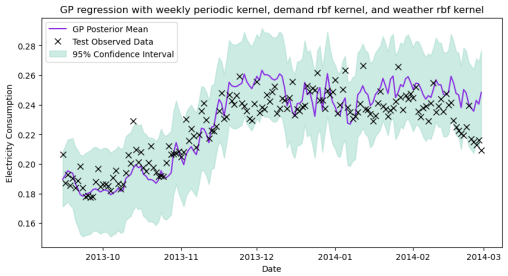
(d) Hyperparameter Tuned: SARIMA(1,1,2)(0,1,2,7) + Weather



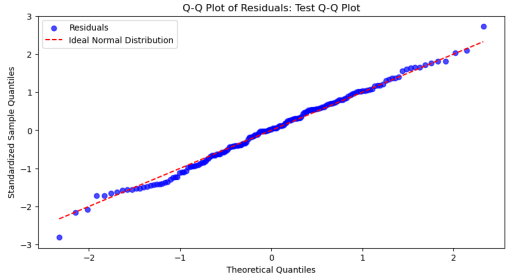
(e) Lower Confidence Intervals



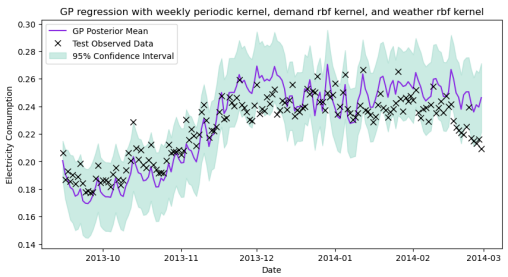
(f) Q-Q Plot for SARIMAX



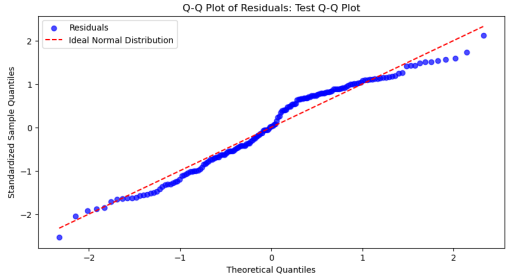
(g) GP using all three kernels with optimize



(h) Q-Q Plot for all three kernels



(i) GP using all three kernels with optimize restarts



(j) Q-Q Plot for three kernels with restarts

Figure 2: Comparison of models

In addition, the fine-tuned SARIMAX model performed extremely well with a MAPE of 2.90 % and continued to track weekly and yearly trends (see Figure 2d. When we reduced our confidence intervals from 95% to 68%, you can see the corresponding narrowing of bands (see Figure 2e).

One thing that was suprising was that at 68% confidence interval, we would expect to see 32% of the test data to fall outside of the interval, but that is not we see here. This led us to check for model misspecifications by looking at the residuals as it should follow a white noise process of zero mean, constant variance and be normally distributed. We checked this using a Q-Q plot (see Figure 2f) and it shows that the residuals adhere to the assumptions of normality. We then checked for calibration error which measures the difference between actual observed frequencies and predicted probabilities. At 68% confidence interval, we have over coverage of 30.80% while at 95%, we have over coverage of 5.00%. In other words, at 95% confidence interval, our model is better calibrated. This makes sense because at wider intervals, it encompasses more data variability. It also means that our training data could have higher variability than the test data which resulted in the model producing predictions with higher standard deviation than necessary and would be something to futher investigate with time.

Gaussian Processes The weekly periodic kernel accurately models seasonality but ignores general demand trends, leading to biased predictions. The demand RBF kernel captures smooth long-term trends but fails to model periodic patterns, resulting in underperformance. Weather data (e.g., temperature, and humidity) have a smooth and continuous influence on electricity demand, but using them alone can cause overfitting, leading to suboptimal results.

Combining the demand RBF and weekly periodic kernels improves the model fit during training by modeling both smooth trends and periodic patterns, but test metrics are bad (high RMSE, MAPE).

Combining all relevant components—the demand RBF kernel, weekly periodic kernel, and weather data—achieves the best performance with accurate and confident predictions. This approach effectively captures smooth trends, periodic patterns, and external influences. (see Figure 2g, 2i)

We also compared the optimization approaches **optimize** and **optimize_restart**. For our best models that consider all three kernels, the **optimize_restart** approach resulted in a better log-likelihood during training, suggesting a superior model fit. However, its predictions on test data showed slightly worse performance in terms of RMSE and MAPE compared to **optimize**. To evaluate model fit further, we used Q-Q plots to visualize the coverage of the 95% confidence interval. The **optimize** approach exhibited overcoverage at approximately 97.6%, while the **optimize_restart** approach showed undercoverage at around 92.8%, both graphs are approximately linear. Despite these differences, the calibration error for both methods was low (approximately 0.026 and 0.022), indicating reliability in interval estimates. (see Figure 2h, 2j)

4 Conclusion

In this project, we learned two key lessons. First, the importance of preprocessing—stationarity and seasonality adjustments were critical in improving the accuracy of our models. Second, the significance of correlated variables—weather data, such as temperature and humidity, enhanced our model performance by capturing external drivers of demand.

Comparison of ARMA models and GP models ARMA models, being parametric and autoregressive, explicitly model relationships between current and past values of the time series, which gives them an advantage in directly capturing recent patterns. However, this reliance on past values introduces increased uncertainty over longer prediction horizons, where errors propagate through the autoregressive process. In contrast, GP models, which are non-parametric and kernel-based, do not rely on explicit past relationships but instead leverage flexible kernels to model complex processes. This flexibility enables GP models to handle more intricate relationships, such as nonlinear trends, multiple seasonalities, or external influences. Furthermore, GP models inherently provide calibrated confidence intervals, offering reliable uncertainty estimates without the compounding effect of prediction errors.

Future Directions Future work could focus on shorter forecasting intervals (e.g., hourly) to provide more actionable insights for energy management and expand the scope beyond London for broader applicability.

Acknowledgment

We would like to thank Professor Sebastian Wagner-Carena for his valuable feedback and guidance on this project throughout the semester.

Fiona contributed to both data exploration and development of ARMA-like models and GP. For the write-up and presentation, she focused on the data, ARMA methods, and their corresponding results and discussion. Patrick contributed to the data exploration and development of GP regression methods. For the write-up and presentation, he focused on the GP methods and the overall learning.

Code availability

Our code is available at Github[3].

References

[1] U.S. Energy Information Administration (EIA). Eia projects nearly 50% increase in world energy use by 2050, led by growth in renewables. Accessed December 12, 2024, n.d.

[2] Bilal Dadanlar. Answer to "multiple seasonality time series analysis in python". Stack Overflow, 2020.

[3] M. Blum and M. Riedmiller. Electricity demand forecasting using gaussian processes. *AAAI Workshop - Technical Report*, 10:10–13, 01 2013.

A ARMA vs auto arima

The parameters chosen by the automated grid search were slightly different at ARIMA(2,1,1). The difference likely arises because the automated grid search is able to consider differencing with AR and MA order terms and optimize for them together, something we are unable to visually pick up when we start considering d = 1 since the data y axis will change and we are not sure how to interpret the ACF and PACF of once differenced data. We will leave a visualization of once differenced ACF (see Figure 5a) and PACF (see Figure 5b) below as a reference of what it looks like. In a similar vein, we plot the theoretical ACF (see Figure 5c) of ARIMA(2,1,1) to compare with the once differenced ACF (estimated ACF) and notice it is not a good fit. Perhaps theoretical ACF of ARIMA(2,1,1) does not work so well in explaining model fit of differenced data.

Additionally, it is interesting to note that AutoARIMA optimizes for AIC. So it focuses on model fit over evaluating forecast performance of unseen data in selecting the best model. In contrast, we prioritized evaluating the model's forecast performance on the test set, using primary metrics such as MAPE, RMSE as our primary goal was to assess predictive accuracy rather than model selection. This is also why AutoARIMA's best model of ARIMA(2,1,1) had worse (higher) MAPE and RMSE than our ARMA(1,0) as it had a better (lower) AIC on train set than ARMA(1,0).

Noticeably, AutoARIMA did pick up on the data set needing to difference once for stationarity which is similar to what we had seen earlier with the ADF test. So this automated grid search is useful in that it allows us to do more when a model requires more parameter selection such as when differencing is involved and for optimizing seasonal parameters.

[3] `https://github.com/chowfi/electricity_demand_forecasting`

B Theoretical ACF

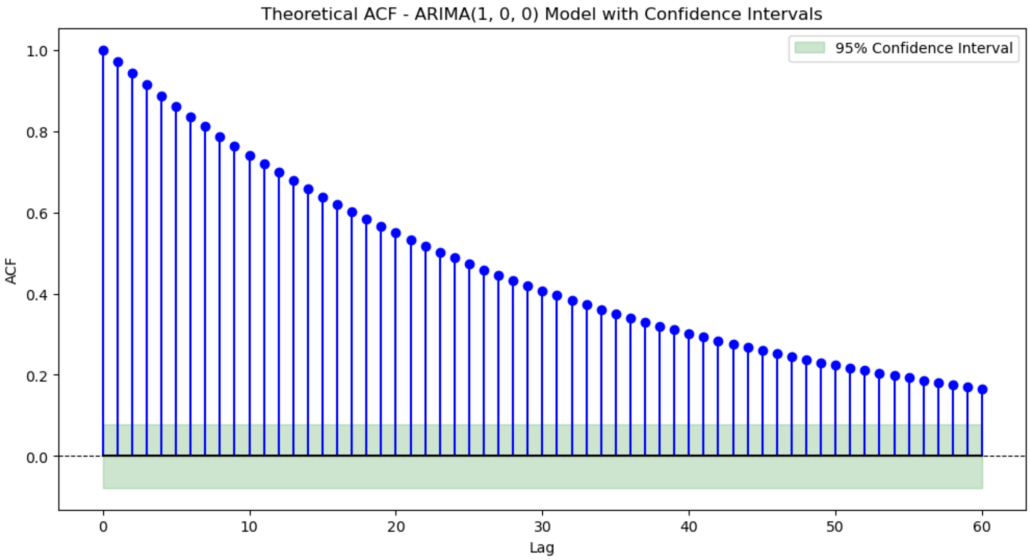


Figure 3: Theoretical ACF of ARMA(1,0)

C Hyperparameter Tuning

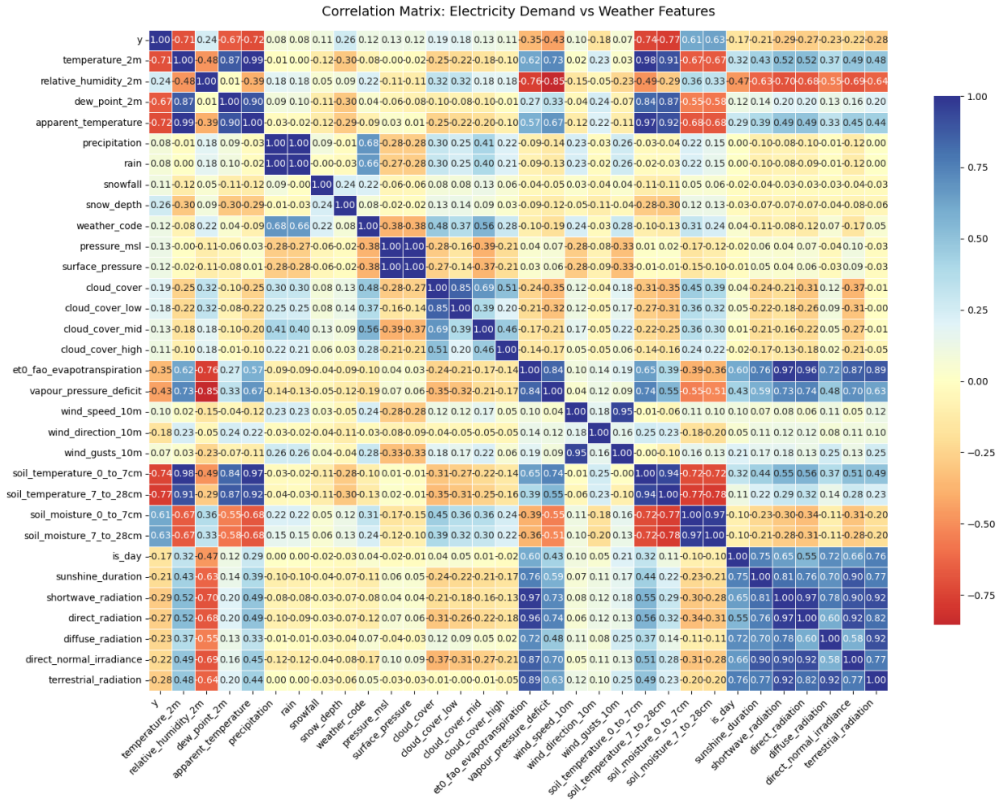


Figure 4: Correlation matrix of electricity demand and weather variables

Table 2: Top 5 Correlated Weather Features to Electricity Demand

Features	Correlation Coefficient
soil_temperature_7_to_28cm	0.7737
soil_temperature_0_to_7cm	0.7413
apparent_temperature	0.7205
temperature_2m	0.7078
dew_point_2m	0.6733

Table 3: Feature Sets Corresponding to Filters

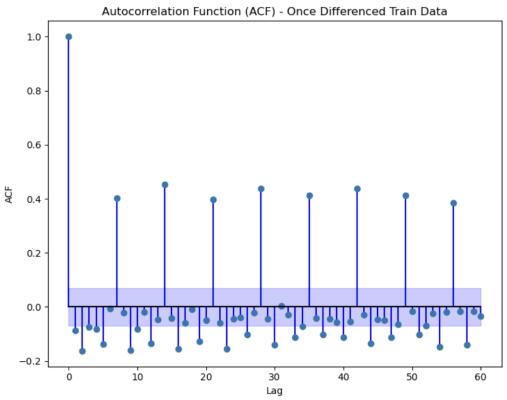
Filter	Features
a	`[soil_temperature_7_to_28cm]`
b	`[soil_temperature_7_to_28cm, soil_temperature_0_to_7cm]`
c	`[soil_temperature_7_to_28cm, soil_temperature_0_to_7cm, apparent_temperature]`
d	`[soil_temperature_7_to_28cm, soil_temperature_0_to_7cm, apparent_temperature, temperature_2m]`
e	`[soil_temperature_7_to_28cm, soil_temperature_0_to_7cm, apparent_temperature, temperature_2m, dew_point_2m]`

Listing 1: Grid search parameter ranges

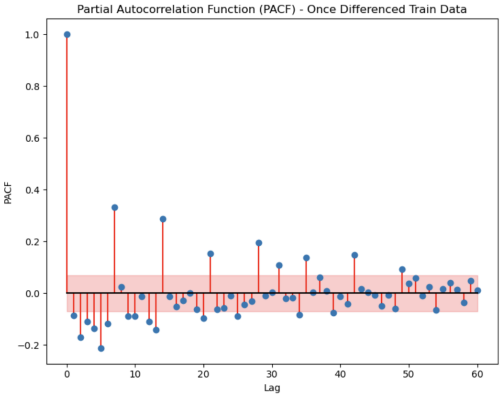
```
p_values = range(0, 4)   # AR order
q_values = range(0, 4)   # MA order
P_values = range(0, 3)   # Seasonal AR order
Q_values = range(0, 3)   # Seasonal MA order
D_values = [0, 1]        # Seasonal differencing order
s_values = [7]           # Seasonal period
```

Table 4: Different combinations of weather variables for exogenous variables in SARIMAX

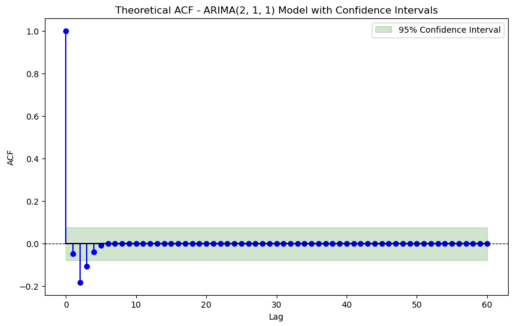
Filter	MAPE	RMSE	Log-Likelihood	AIC
a	0.04050	0.01105	2306	-4596
b	0.03845	0.01069	2322	-4627
c	0.03704	0.01035	2322	-4625
d	0.03563	0.00981	2330	-4638
e	0.07562	0.01946	2343	-4662



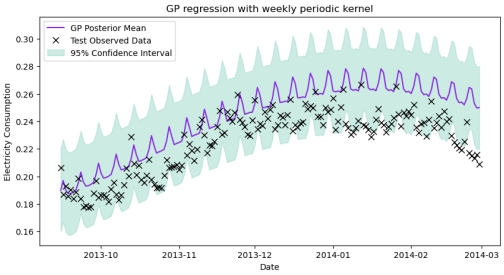
(a) ACF – once differenced



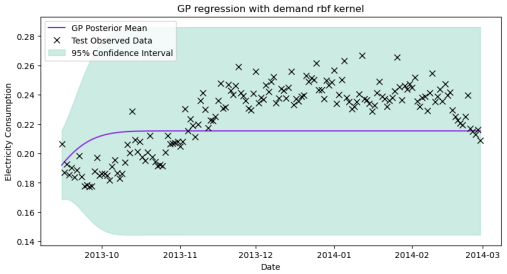
(b) PACF – once differenced



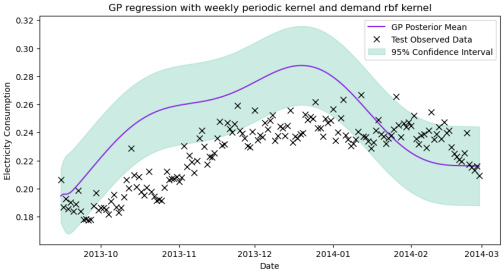
(c) Theoretical ACF of ARIMA(2,1,1)



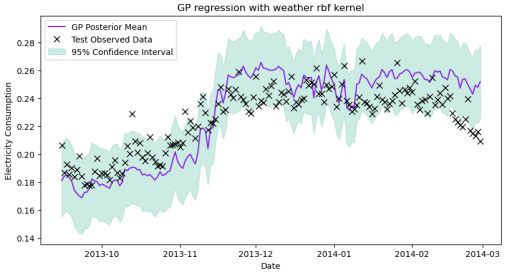
(d) Weekly Periodic Kernel Only



(e) Demand RBF Kernel Only



(f) Periodic Kernel + RBF Kernel



(g) Weather RBF kernel