BOSTON UNIVERSITY

ARIMA-Based IT Power Forecasting

Sophia Holland[1,2], Daniel Wilson[2], Professor Ayse Coskun[2]

Lausanne Collegiate School, Memphis, TN [1], Boston University Electrical and Computer Engineering Department, Boston, MA[2]

Introduction

In a Smart Grid system, electricity providers determine power demand from energy users and match their supply accordingly. This saves power and reduces costs for all parties involved.

Users can participate in the Smart Grid by giving electricity providers a bid for how much power they will use for a given amount of time in the future, allowing providers to adjust how much power gets generated, stored, and distributed. Having a model that can accurately predict a user's power demand is a key part of this process.

Using data from the Massachusetts Green High Performance Computing Center (MGHPCC), an academic data center, I have constructed an ARIMA time series forecasting model to generate out-of-sample IT (computing/non-cooling) power predictions.

Preprocessing

Resampling:

Every 8 hours (4 data points per day)

timestamp	it-power
2018-01-06 00:00:00.000000	9.480807e+05
2018-01-06 05:05:08.959775	9.566203e+05
2018-01-06 05:05:31.089097	9.566306e+05
2018-01-06 05:05:53.981808	9.566306e+05
2018-01-06 08:00:00.000000	9.574011e+05

Above: Data Before Resampling

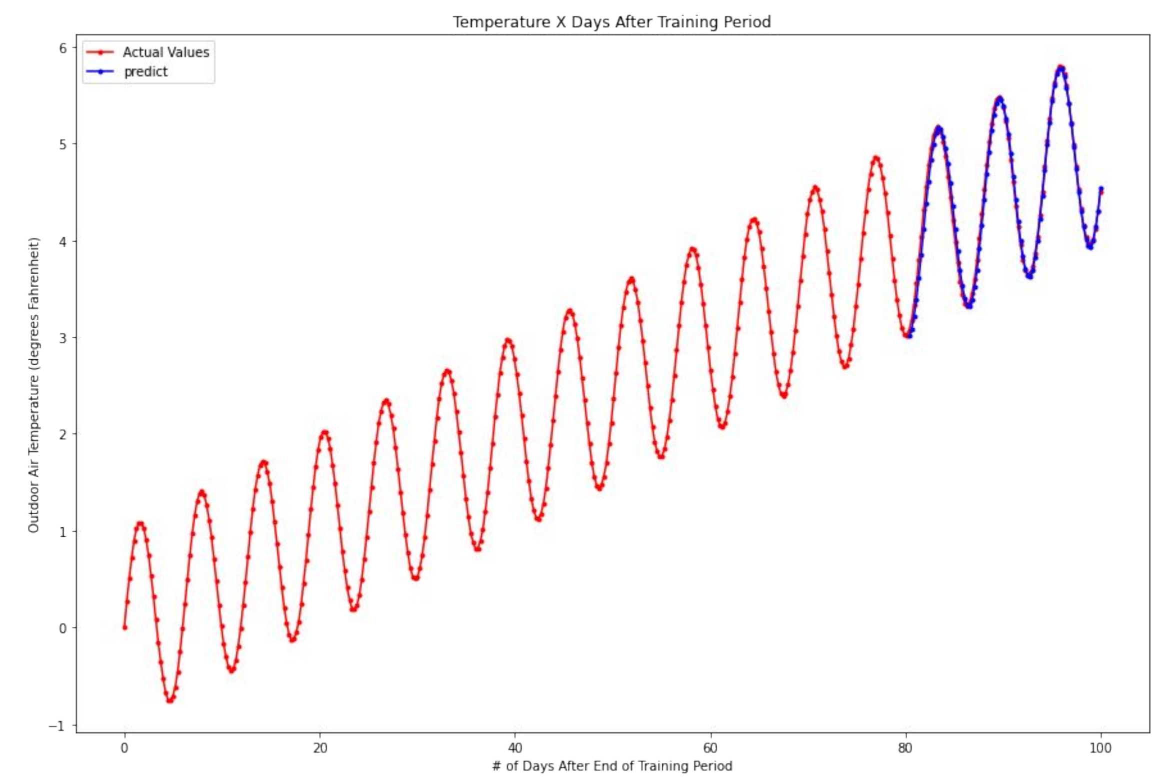
timestamp	it-power
2018-01-06 00:00:00	9.480807e+05
2018-01-06 08:00:00	9.574011e+05
2018-01-06 16:00:00	9.527241e+05
2018-01-07 00:00:00	9.360466e+05
2018-01-07 08:00:00	9.188817e+05

Above: Data After Resampling

Interpolation:

Missing Values are an average of their two closest neighboring values.

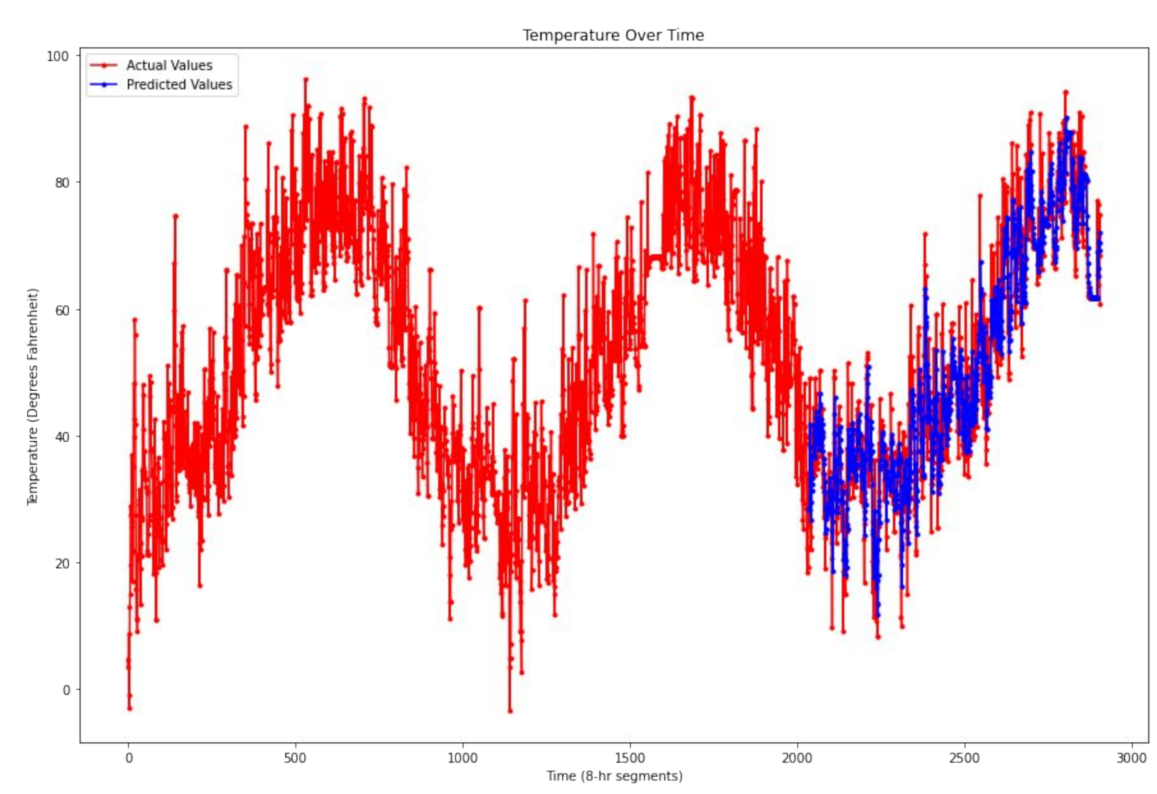
Results



Sine Wave Prediction

RMSE: 0.00652 Order: (1, 0, 1)

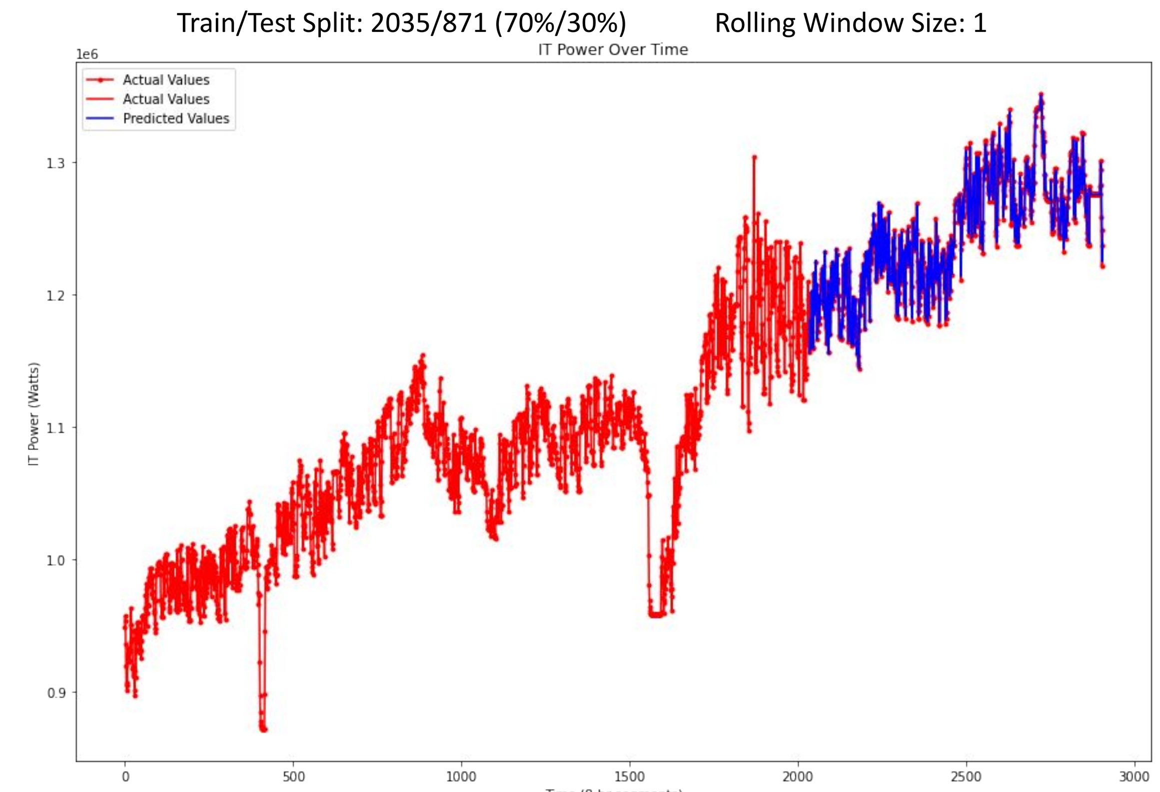
Train/Test Split: 324/80 (80%/20%) No Rolling Window



Outdoor Air Temperature Prediction

RMSE: 47.3 Order: (1, 1, 2)

Train/Test Split: 2035/871 (70%/30%) Rolling Window Size: 1



IT Power Prediction

RMSE: 16677.88 Order: (1, 1, 1)

Train/Test Split: 2035/871 (70%/30%) Rolling Window Size: 1

Discussion/ Conclusions

Improving Overall Model Strength

Used three datasets to develop a robust model:

Sine Wave

- Seasonality + linear trend
- Regular, predictable patterns

Outdoor Air Temperature Data

- Noisy data + seasonality
- Patterns can be estimated visually

IT Power Data

- No clear seasonality
- Noisy data + linear trend

Rolling Window Approach

By predicting only one future value at a time, the forecasts became more accurate than when many forecasts were made at once. This sort of continuously rolling approach should make the model more robust when handling new data over a long period of time; It was a key factor in the model's success.

Error Calculations

Root Mean Square Error (RMSE) was the method used to determine the model's deviations from the real values in the dataset. It can be expressed as the equation

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

(Holmes)

RMSE is the standard deviation of the difference between the predictions and real values. A perfect model would have an RMSE of zero.

I found that less seasonal data showed a greater RMSE than data with regular seasonality.

Overall Prediction Trends

Predicted values tended to stay closer to the moving average in noisy datasets, rarely matching the highest and lowest values. This could potentially by improved by increasing the *p* value, which would greatly increase the amount of time the model needs to run.

Methods

ARIMA Model

(AutoRegressive Integrated Moving Average)

- Model commonly used to produce time series forecasts
- Found in the Statsmodels library for Python
- Train/Test split must not be randomized since time sequence matters in pattern detection

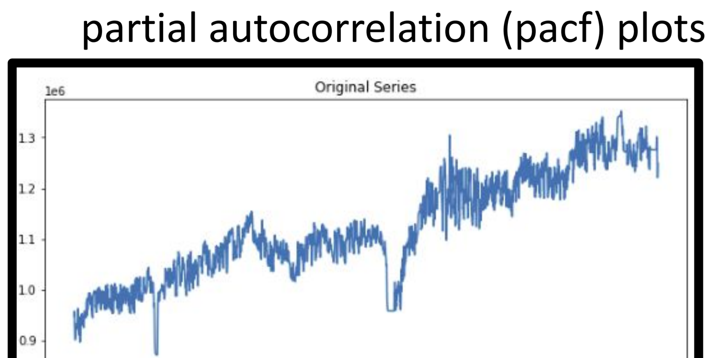
(Brownlee)

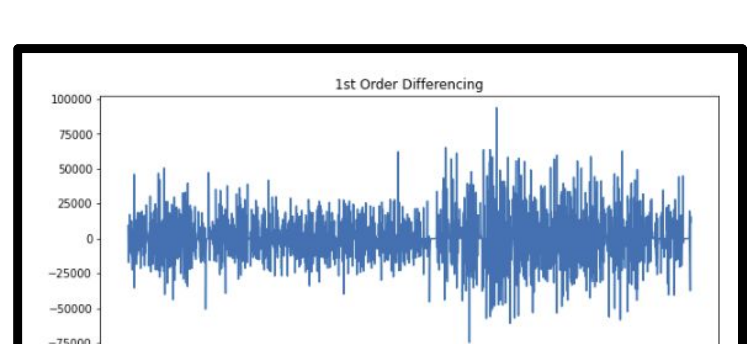
3 variables make up the Order: (p, d, q)

p = lag order (number of lags; determined by correlation between lags and present values) ⟶ determined from autocorrelation (acf) plots

d = order of differencing

q = moving average order (size of averaging window) ⟶ determined from partial autocorrelation (pacf) plots





Original IT Power Series vs IT Power Differenced Once

Proper differencing gives the data a constant overall mean, used as a baseline from which variations can be predicted.

References

Brownlee, Jason. How to Create an ARIMA Model for Time Series Forecasting in Python https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/ (Accessed Aug 2021).

Holmes, S. RMS Error https://statweb.stanford.edu/~susan/courses/s60/split/node60.html (Accessed Aug 2021).

Acknowledgements

Many thanks to Daniel Wilson, Professor Ayse Coskun, and the PEACLab team at Boston University for their instruction and support throughout this project.