Prediction of Electricity Draw and Net Metering

Liam Golly
Department of Electrical Engineering
University of California, San Diego
lgolly@ucsd.edu

Srujan Gutta
Department of Data Science
University of California, San Diego
sgutta@ucsd.edu

Patrick Yeh
Department of Computer Science
University of California, San Diego
psyeh@ucsd.edu

Abstract

Recent changes to California laws with the passing of legislation surrounding NEM 3.0 have negatively impacted many of the incentives supporting the solar industry. As a result, it is necessary for the industry to find methods to optimize profit margins, and predicting electricity draw is a common strategy. However, common methods currently employed are still plagued by issues such as inaccuracy. As a result, this study aims to examine the performance of utilizing deep neural networks for predicting electricity draw. In particular, we evaluate the performance over an assortment of architectures, including several variations of the LSTM model. From our experiments, we identified the unidirectional convolutional LSTM as the best model for our task, though the results suggest that further refinement on this approach is needed improve electricity draw - and, ultimately, peak hour - predictions.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

1 Introduction

With the introduction of the updated NEM 3.0 law in California [2], citizens who are interested in investing in solar must contend with the fact that the reward for putting their generated power back into the electricity grid, also known as net metering, is now derived as a function of electricity draw, or the amount of power currently being used. The implications are that homeowners will now no longer be paid a fixed rate for their solar energy, meaning it will take longer for them to recoup their investment for solar panel installation. Due to a lack of research into predicting electricity usage based on average consumer use across the grid, and an increased need for knowledge following the changes to state law, we decided to employ a deep neural network with the goal of predicting electricity draw based on a variety of factors. Such results would:

- Help homeowners make informed decisions about the optimal time to begin dumping energy back into the grid
- Promote widespread solar adoption by improving return-on-investment which has implications on climate change mitigation
- Advance research into the use of solar power in California
- Further knowledge into the effectiveness of neural networks in applications related to predicting the consumption and production of electricity

In particular, having more accurate predictions is crucial for cases where the ideal window to dump power into the grid may be over a period of only one to two hours. We believe that the usage of external data such as location, temperature, weather conditions, and other additional factors may lead to more accurate, detailed, and personalized responses for anybody who may find use from this data. Because applying neural networks to predict draw is a relatively recent endeavor, our work was exploratory by design as we analyzed new methods of predicting electricity draw based on a number of the aforementioned factors, in order to identify which features were most pertinent to our prediction task.

2 Related works

Currently, the EIA conducts most of the advanced research for predicting electricity draw on the power grid, with the results published on their website.[1] However, their conclusions are solely based on the corresponding electricity draw of the previous year. This information, while generally useful for capturing the similarities these trends have to those of previous years, fails to be the most accurate source. Other researchers, such as Soon,[8] have predicted electricity draw for particular locations or buildings. However, few researchers until very recently have looked into prediction models targeted towards consumers. At the time of writing, Irfan[4] has recently published a paper on a similar topic to ours. Irfan uses similar models to ours, and arrives to the same conclusion regarding the efficacy of LSTM models.

3 Methods and approach

We wanted to answer whether training models could be adopted to accurately model the electric draw of the grid over time, using this model to ultimately help predict the peak hour. Towards this end, we performed extensive data preprocessing, trained a collection of different models, and used RMSE to evaluate the performance of these models.

3.1 Data processing

For the core section of our data, we obtained timestamped electricity draw values on every subgrid in California over the past four years. This formed the backbone of our data and was accordingly used in every test. By nature, the data was also noisy, and by itself difficult to train on due to an abundance of outliers and circumstances that were difficult to predict. One can see the inherent noise of this data through graphs, as shown below.

One major example of this inherent randomness we found was the 2019 wildfire season in central California. During this time, many electricity distributors intentionally went into a self-induced

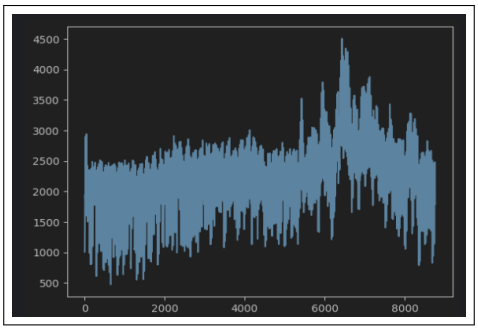

Figure 1: Electricity draw over the second half of the 2019 year and the first half of the 2020 year.

blackout in order to minimize the risk of starting or otherwise exacerbating wildfires.[6] This cause a dramatic dip in power during the summer months, times where, in this region, one would expect to see heightened electricity draw due to the use of air conditioning and the influx of children staying at home during the summer months. Because it was impossible to filter out every outlier in this dataset, we decided to use the dataset as-is, and to mitigate the noise we incorporated other features with the hope that these models would be able to correlate and account for the noise that we saw. Additionally, some other data that we believed would be useful to our prediction task was data pertaining to weather and atmospheric conditions. To test what data might be correlated, we brought in over 40 unique features to test from an open source historical weather and atmospheric collective. [7] Due to the fact that the data acquired from the EIA and from the historical weather database did not line up in location or time, a significant amount of preprocessing was exercised to ensure that the data reflected what we believed to be the proper conditions at the correct time and location. However, the historical data site was somewhat faulty, and many features were thus unusable. Our primary focus was on using temperature, cloud cover, and weather. However, we believe that there are more features that could be used to produce better results.

In another effort to normalize and fix some of the noise that we found in the dataset obtained from the EIA, we used Matlab to take the fast Fourier transform (FFT) of the electricity draw dataset, then fully removed all frequencies that were below a certain threshold. This had noticeable effects, as shown in the following image:

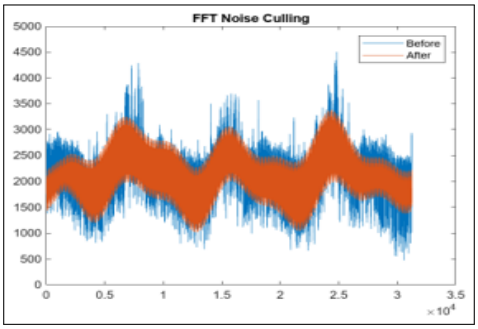

Figure 2: A before and after on using an FFT to cull noise.

This method, while certainly useful for getting rid of noise, also eliminates many of the finer details, and even when we obtained a lower error while training on this data, we still find that the model doesn't retain its effective predictive capabilities when put into real scenarios, much to our dismay.

3.2 Models

For our experiments, we tested on a variety of models to ascertain their effectiveness with our dataset. The following section is aimed at introducing the major details of these models. Refer to Table 1 in results for the exact times and RMSE for all of these models.

3.2.1 Prophet

Facebook Prophet is an open source library developed by Facebook for performing time series forecasting. [5] Prophet is designed to handle the common features of business time series data with trends, seasonality, and holidays. That was one of the main reasons why we chose to try to use Prophet since our data could very well be affected by trends and seasonality. The other thing we hoped it would help with was preventing the effects of outliers. We had to do a couple of preprocessing steps to ensure the model would become easy to use. To do this, we one-hot-encoded our categorical columns and split our time column into 4 different columns - year, month, day and hour - to potentially help Prophet make better predictions. After the preprocessing steps were completed, implementing the model became quite simple since we could directly call the Prophet model by importing it. The results of the Prophet model were unfortunately not as great as we hoped with the error being an RMSE of nearly 8000.

3.2.2 ARIMA

ARIMA is AutoRegressive Integrated Moving Average, which is used for forecasting time series data. "AR" is the past values the model uses to predict the future values, and "I" stands for the potential difference between past values and the predicted to make the time series stationary. "MA" uses the past errors of how wrong the past predictions were to make better future predictions. We did the same preprocessing steps for ARIMA as we did for Prophet. Once again, the advantage for this model was the implementation time and speed at which it runs. The downside, unfortunately, was that the accuracy still remained too high to be useful with an RMSE around 4500.

3.2.3 SARIMA

Since our data most likely exhibits seasonal patterns, we decided to use Seasonal AutoRegressive Integrated Moving Average or SARIMA [3] to incorporate a seasonal component to the data. This approach promised to be useful since certain seasons have weather patterns that can lead to people using more power. Unfortunately, this model also had a high RMSE and actually ended up performing worse on our dataset than ARIMA. However, once again, one of the best benefits of using this model was its speed and ease of implementation.

3.2.4 XGBoost

XGBoost or eXtreme Gradient Boosting users boosted tree algorithms, it is known for speed and performance. The real benefit we were looking at for using XGBoost was that it can handle overfitting really well, we thought because of the amount of outliers in our data that this may help. Unfortunately while this had gotten our best RMSE in the fastest time it also had an RMSE of near 3700 which was still far too high.

3.2.5 LSTM

LSTM (Long Short-Term Memory) is a well-understood deep neural network model that excels in finding patterns in periodic and semi-periodic data with a wide range of frequencies. This advantage is due to its ability to capture both long- and short-term patterns. Due to our experience with LSTM and its known success for similar problems and datasets, we believed that this model would see a decent amount of success. To implement LSTM, we mostly focused on ensuring that the input data was well-formatted and usable, and that our hyper-parameters were well-tuned to our dataset. Similarly to our previous models, we split our data into multiple features in order to help the model find associations with the different periods of time. We also processed and fixed the weather data to be compatible with the model. After tuning the model, we found that the LSTM provided better results than previous tests. However, these results were still of middling accuracy, and we believed that further improvement could be made.

3.2.6 Unidirectional Convolutional LSTM

A Unidirectional Convolutional LSTM is a variant of the recurrent neural network architecture that is especially designed to handle sequences of data with long-range dependencies. Here, the term "unidirectional" implies that the LSTM only processes the data from the past to predict future values.

The Convolutional LSTM (ConvLSTM) model was proposed to deal with spatio-temporal data, which involves both spatial and temporal dimensions. Instead of using fully connected operations as in the original LSTM, ConvLSTM uses convolutional operations in its input-to-state and state-to-state transitions. In terms of performance, this was by far the best model with an initial validation RMSE of 892. Upon this realization, we began to conduct extensive hyperparameter tuning with optuna and created study trials. Another optimization we implemented was a learning rate scheduler, which helped drop our RMSE further after we hit a plateau and we got it down to 289.

3.2.7 Bidirectional Convolutional LSTM

The Bidirectional Convolutional LSTM is similar to the Unidirectional Convolutional LSTM, except that the bidirectional form of the model means that the LSTM reads the input sequence from both directions (past-to-future and future-to-past), which can provide additional context compared to a Unidirectional LSTM. This model unfortunately took more time to run and seemed to overfit to the data as we witnessed greater variance in the validation loss in comparison to the Unidirectional Convolutional LSTM.

4 Results

After testing these models, we found the following:

Table 1: Results and info

Model Analysis		
Model	Validation RMSE	Training Time (seconds)
ARIMA	4578	22
SARIMA	4737	211
Prophet	6552	28
XGBoost	3677	0.5
LSTM	2576	272
UCL	289	316
BCL	415	381

After running tests with different models and preprocessing approaches, we found that, while model choice was extremely impactful, it was also extremely important for us to use advanced data processing techniques in order to obtain the best results. Our final dataset included multiple features related to the time, location, with various features related to the weather and atmospheric conditions. We found that the best training occurred with a train-test split ratio of 0.8 to 0.2, respectively, where we train on a random sample of 10,000 data points at any given iteration. When testing hyperparameters, we found the best hyperparameters to be the following:

Table 2: Hyperparameters

Unidirectional Convolutional LSTM Parameters					
Epochs	Learning Rate (Max)	Filters	Kernel Size	Hidden Dimensions	LSTM Layers
400	0.001	64	3	128	2

Even with a suitable model choice and design, such as the unidirectional convolutional LSTM with a dynamic decaying learning rate, we still weren't able to achieve as low a RMSE as when we used a simpler model with a more processed data set. It was only when we combined the methods that we achieved results that were beginning to fall within our expectations.

Overall, we found that our results aligned with the conclusions found by Irfan,[4] as we both attempted to make use of similar models during our research and found that we both had similar flaws before moving on. Our project differs in that our main focus was not to ultimately predict the electricity draw on the grid, although this was a beneficial side effect. Rather, we aimed to predict when was the peak time for putting solar-generated electricity back into the grid for maximal returns. Because

of this, we believe that future work could be optimized further for those who wish to continue our research into this specific application.

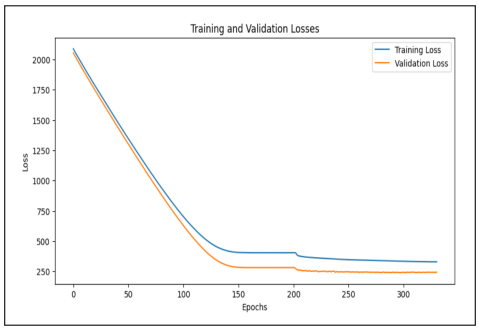

Figure 3: Results from the UCL with advanced data preprocessing.

To reproduce our results, the code can be found at:
`https://colab.research.google.com/drive/1ol4TzCLO3RPc-oQBjmq0Va4dBDNE7P5v?usp=sharing`

5 Future work

Though the initial results are quite promising, there are several potential modifications and improvements in our approach and implementation that can further expand on our discoveries. For instance, utilizing better hardware and finding more optimal algorithms (and better implementations of these algorithms) can improve training efficiency and limit some of the issues regarding memory during the training steps. This could make our findings more tenable for adoption by those with a vested interest in solar, such as energy providers or perhaps even individual consumers. Furthermore, augmenting our data with features such as the specific location could convey important information that is essential to accurate predictions: it is fair to assume that locations near electricity distributors, such as San Diego, would have markedly different behavior patterns compared to more remote locations such as the outskirts of Northern California. Finally, directly predicting peak hour, rather than using electricity draw as a proxy, is an enticing possibility that further studies can research. Indeed, that is the true problem at hand. However, due to the high variance and erratic nature of the data, we would need to first find and experiment with further techniques that deal with this noise. For example, utilizing the FFT algorithm was one approach used to simplify the data, and using this transformed data we can examine the periodic nature of the data to predict when peaks occur, but other strategies certainly merit some consideration and further study is needed.

Of course, another major avenue to explore is the use of different training architectures. Though this study considered a few different models, such as LSTMs, unidirectional convolutional LSTMs, and bidirectional convolutional LSTMs, additional research may be taken to analyze the performance for an even greater variety of models with respect to predicting peak hour. Indeed, as further advancements within the field of machine learning emerge and develop, it is both prudent and necessary to evaluate their performance within applications such as net metering.

6 Individual contributions

LG, SG, and PY were involved in the original formulation, experimental design, and data analysis of the project. LG and SG worked on preprocessing the data, implementing the model architectures, as well as running the experiments. LG, SG, and PY all collaborated in conducting the literature review, creating the various figures and tables for the report, and writing out the research paper.

References

[1] *Energy Information Administration*. 2023. URL: https://www.eia.gov/electricity/data/browser/.

[2] California State Gov. *Proposed Decision Revising Net Energy Metering Tariff and Subtariffs*. 2022. URL: https://docs.cpuc.ca.gov/PublishedDocs/Published/G000/M499/K921/499921246.PDF.

[3] Jamie Halliday and Georgi N. Boshnakov. *Partial autocorrelation parameterisation of models with unit roots on the unit circle*. 2022. arXiv: 2208.05055 [stat.ME].

[4] Muhammad Irfan et al. *Multi-region electricity demand prediction with ensemble deep neural networks*. 2023. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0285456.

[5] Meta. *Prophet*. 2023. URL: https://facebook.github.io/prophet/.

[6] Emma Newburger. *More than 2 million people expected to lose power in PGE blackout as California wildfires rage*. 2019. URL: https://www.cnbc.com/2019/10/26/pge-will-shut-off-power-to-940000-customers-in-northern-california-to-reduce-wildfire-risk.html.

[7] *Open Weather API*. 2023. URL: https://open-meteo.com/.

[8] Chua Chiah Soon et al. *Predicting energy demand with neural networks*. 2020. URL: https://towardsdatascience.com/forecasting-energy-consumption-using-neural-networks-xgboost-2032b6e6f7e2.