COVID-19 Reduces Electricity Demand in U.S. Cities

Deep Dayaramani
Dept. of Civil and Environmental Engineering
Stanford University
Stanford, CA, USA
deepdaya@stanford.edu

Joshua Geiser
Dept. of Aeronautics and Astronautics
Stanford University
Stanford, CA, USA
jkgeiser@stanford.edu

Fletcher Passow
Dept. of Statistics
Stanford University
Stanford, CA, USA
passow@stanford.edu

I. Introduction

Life as Americans know it depends on a reliable electric grid. People need electricity to preserve food, pump water, wash clothes, and operate increasingly digital businesses. The COVID-19 pandemic, and the public health policies that accompany it, have changed the way that people consume electricity by radically altering people's way of life. People are spending less time in public and more time at home. Many individuals able to work from home may not have visited their offices in months. Trips to stores, even for essential products, may be less frequent. Changes like these, aggregated over all the inhabitants of a large city, can significantly impact the behavior of that city's entire electrical infrastructure. Electric grid operators and energy policy-makers need to understand the impact of extreme events, like pandemics, on the electric grid so that they can maintain the grid's reliability today and prepare for disruptions in the future.

Quantifying the impact of the pandemic on a city's electricity demand is difficult. Electricity demand depends on many factors, including weather, the day of the week, and the time of the day, just to name a few [1] [2]. Simply comparing demand across years without adjusting for variations in these factors can lead to erroneous conclusions. To carefully assess the pandemic's impact, we needed to compare observed electricity demands with an estimate of the unknown demand that would have occurred absent the pandemic, also known as the "counterfactual" demand.

In this project, we approached the problem in two stages. First, we created a long short-term memory (LSTM) neural network algorithm to predict counterfactual demands [3] [4]. The algorithm predicted electricity demand in megawatts (MW) for a given hour in a given city. The inputs to the algorithm were hourly weather data for the given city and calendar-related features (e.g. weekday, weekend day, hour of day). We trained, validated, and tested the model on pre-pandemic historical data. Second, we used the algorithm to predict the counterfactual electricity demand each city might have had absent the pandemic. Applying the algorithm hour-by-hour to time series of input data for each city gave us counterfactual electricity demand time series for each city. Finally, we drew conclusions about the impacts of the pandemic on electricity demand by comparing our counterfactual scenario to the observed electricity demands.

II. Related Work

Several previous studies assess the impacts of the COVID-19 pandemic on electricity demand [5] [6] [7]. We organize them here in order of increasing model complexity and dataset size. Bahmanyar, Estebsari, and Ernst considered changes in electricity demand across six European countries. Their study quantified impact by comparing electricity demand for a week at the beginning of the pandemic (the second week of April 2020) to demand for a similar reference week from 2019. They found that electricity demand in different countries changed very differently. For example, they found that Spain's electricity demand decreased by 25% while Sweden's demand increased by 2% [5]. Agdas and Barooah evaluated changes in electricity demand for the city of Gainesville, Florida. They modeled electricity demand using an OLS linear regression. To train the model, they used four weeks of data from March 2019. They found that electricity demand in Gainesville increased by 10% (95% CI: 2% to 16%) in 2020 compared to 2019 when corrected for temperature and time-of-week effects [7]. Prol and O estimated changes in electricity demand for several E.U. countries and large U.S. states. They trained their model, a dynamic harmonic regression, to predict daily electricity demand based on data from January 2015 to July 2020 [8] [6]. They found that "the cumulative decline in electricity consumption within the 5 months following the stay-home orders ranges between 3% and 12% in the most affected EU countries and USA states" [6].

None of these studies used machine learning algorithms. However, the electricity industry uses machine learning in daily operations to predict demand. According to Hippert et al., thirty U.S. electric utilities were using forecasting algorithms based on a particular neural network in 1998 [9]. Hahn, Meyer-Nieberg, and Pickl observe that traditional feed-forward neural networks are the most often applied type of neural network. However, they also mention the use of radial basis function networks, self-organizing maps, and recurrent neural networks [2]. Since 2009, when Hahn, Meyer-Nieberg, and Pickl published their review, the machine learning field has advanced considerably. In this project, We sought to improve on the results of the pandemic impact studies listed previously by using more modern algorithms. In particular, we chose to build a recurrent neural network based on the long short-term memory (LSTM) algorithm [3] [4].

III. Dataset and Features

Although the previous studies of COVID-19's impacts on electricity demand used different models to predict electricity demand, they all selected similar input variables. All three studies mentioned previously recognized the critical roles that weather and calendar effects play in driving electricity demand [5] [6] [7]. For this research project, we chose the **COVID-EMDA+** data hub as the data source [10]. This data set consisted of hourly weather, load, and COVID-19 data for major U.S. cities and regions from January 2017 - September 2020. The raw data set was fairly clean, however some pre-processing work was required to transform the raw data into a more usable hourly format.

A summary of the seasonal load trends for select U.S. cities can be seen in Figure 1. This figure suggests that date plays an important role in predicting load magnitude. For example, each city appears to have spikes in load over summer months. Looking on a shorter time scale, we also noticed oscillatory load behavior over the course of a day, which peak around the mid-afternoon and evening hours. From this and other exploratory data analysis, raw features including date, temperature, and time of day emerged as strong predictors of load patterns.

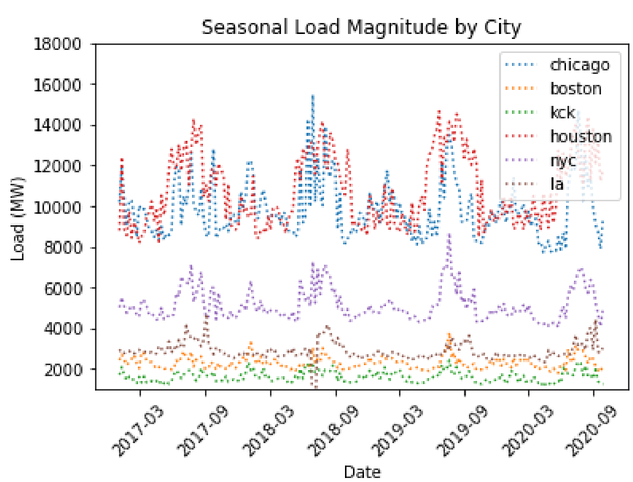

Fig. 1. Seasonal Load Trends for 6 Major U.S. Cities

From these raw features, we engineered various other features from our data set. To see how our variable depended on the day, we extracted weekday, weekend, pre-weekday, pre-weekend and holiday features from our data-set. To see the correlation with respect to weather, we evaluated the cumulative and exponential moving averages for each of the weather variables along with the maximum, minimum and mean day wise for the previous 24 hours.

After generating these features, we split our dataset into four components for training, validation, testing, and COVID-19 impact evaluation, respectively. Each component included data from five cities (Boston, Chicago, Houston, Kansas City, New York City) over date/time ranges as specified in Table I. For models trained on a single city, we arbitrarily chose to focus on Houston.

TABLE I
DATASET SPLITTING BY DATE RANGE, WITH TOTAL NUMBER OF SAMPLES FROM ALL CITIES

Dataset	Date Range	# of Samples (5 cities)
Train	Jan. 2017 - Sep. 2019	119,525
Validation	Oct. 2019 - Dec. 2019	11,035
Test	Jan. 2020 - Feb. 2020	7,200
COVID Comparison	Mar. 2020 - Sep. 2020	25,560

IV. Methods

Before implementing an LSTM neural network, we created two baseline models. These models gave us a better understanding of the relationships in our data. They also revealed the flaws in assuming a linear relationship between electricity demand and our predictors.

A. Baseline Models

1) Ordinary Linear Regression: A preliminary model was generated using Houston feature/load data between January 2017 and September 2019 using a simple linear regression model. This model was used to analyze the effect of weather and time features on the load. The hypothesis function $y(t)$ and loss function $J(\theta)$ are shown below.

$$y(t) = \theta^T x(t) + \epsilon_t$$
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - \theta^T x^{(i)})^2$$

where $\epsilon_t = \mathcal{N}(0, \sigma^2)$, is a standard normal error.

2) Autoregression with Prophet: Another preliminary model was generated using Auto-regression to understand load models using only the load variable. This model was used to understand the dependence of the load variable on only past observations. For this, implementation of auto-regression in the package-*FB Prophet* was used. A MAPE is shown in the results section. The hypothesis function, fitting function in the form of maximum a *posteriori* estimate are shown below:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$
$$y \sim \mathcal{N}((k + A * \delta) \odot t + (m + A * \gamma) + X\beta, \sigma)$$
$$k, m \sim \mathcal{N}(0, 5), \epsilon \sim \mathcal{N}(0, 0.5), \delta \sim \mathcal{L}(0, \tau), \gamma \sim \mathcal{N}(0, \sigma)$$

Here, $g(t)$ is the trend function which is allowed with change-points, $s(t)$ captures seasonality of any form, and $h(t)$ is the holiday function. Also, σ is the model variance, A represents the changepoints $a(t)$, points where trend growth rate can change, and δ represents the trend rate approximated by a Laplace distribution. [11]

B. Advanced Models- Long Short Term Memory Recurrent Neural Networks (LSTM-RNN)

RNN's are used to learn temporal variations and patterns in various types of data. But RNN's suffer from Short Term Memory, because of the vanishing gradient problem, one which limits the learning in most earlier states with small gradient updates. LSTM's were proposed as a solution to this problem. The model adds gates and cell states in order to retain longer sequence patterns in RNNs (see Figure 2 from [12]).

TABLE II
LSTM TERMINOLOGY AND VARIABLES

Variable	Name
$f(t)$	Forget Gate
$\sigma(x)$	Sigmoid Function
h_{t-1}	Hidden Data at time-step t-1
x_t	Input Data at time-step t
i_t	Input Gate
o_t	Output Gate
c_t	Cell State at time t
c_{t-1}	Cell State at time t-1
h_t	Hidden Data at time t-1
W	Weight Matrices
b	Bias Vectors

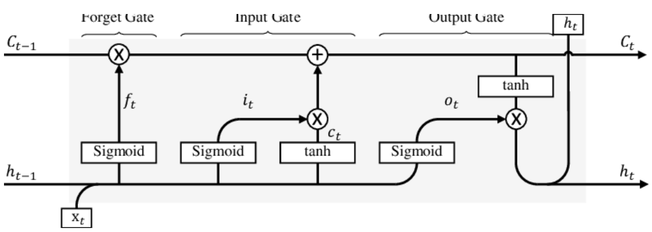

Fig. 2. LSTM Memory Block with Operations

Forget Gate determines the amount of data to forget, Input Gate determines which data to store, and Output Gate determines the information to let out. The cell state carries relevant information throughout the sequence, so it can store pieces of information over multiple time steps, hence reducing the impact of short term memory [13].

$$
\begin{aligned}
&f_t = \sigma(W_{f_{xh}} + W_{f_{hh}} h_{t-1} + b_{f_h}) \\
&i_t = \sigma(W_{i_x h} x_t + W_{i_h h} h_{t-1} + b_{i_h}) \\
&o_t = \sigma(W_{o_{xh}} x_t + W_{o_{hh}} h_{t-1} + b_{o_h}) \\
&g_t = \tanh(W_{g_{xh}} x_t + W_{g_{hh}} h_{t-1} + b_{g_h}) \\
&c_t = f_t \odot c_{t-1} + i_t \odot g_t \\
&h_t = o_t \odot tanh(c_t)
\end{aligned}
$$

For this project, we chose to use an LSTM because it is well adapted to time series applications, according to the advice of a course TA.

To feed data in the form of tensors into our models, we had to choose a "window-size", which corresponds to the number of samples of data in each slice of the tensor. Each slice corresponds to a specified number of time steps of input and the output is a single time step.

The concept of hidden units in an LSTM corresponds to the dimensionality of the outputs of each gate. Our outputs would be (hidden_units, 1) size.

We implemented our LSTM in Tensorflow using the Keras interface. [14] [15].

V. EXPERIMENTS AND RESULTS

We used four main strategies to improve our LSTM model predictions. The first was engineering new features to compensate for visible prediction errors. The second was a hyperparameter tuning search. The third strategy was validation and training error versus training set size plots as described in Lecture 16 [16]. The fourth was experimentation with the number and type of layers included in the model.

All strategies used the same error metrics to guide them. Hahn, Meyer-Nieberg, and Pickl observed that previous studies of electricity demand prediction frequently use Mean Absolute Percentage Error (MAPE) [2]. We follow that convention in this report, calculating MAPE as follows:

$$MAPE = \frac{100}{T} \sum_{t=1}^{T} \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

Here, T represents the total number of time steps (samples), y_t represents the observed electricity demand at time t, and $\hat{y}_t$ represents the predicted electricity demand at time t.

The first improvement strategy was engineering features to compensate for visible prediction errors. We used graphs similar to Figure 4 to examine each model's performance on the validation dataset. Each feature was also normalized to a mean of zero and standard deviation of one, such that no one feature would dominate the training. After some initial experimentation with the LSTM, it appeared that the model tended to consistently underpredict or overpredict load at various intervals throughout the year. This suggested that certain seasonal trends were not being accounted for by our model. To account for this, we added an additional week-of-year feature to our dataset to account for this seasonal variability.

It was also noted that the model generally was able to capture the oscillatory behavior of the true load, however it frequently overshot the "peaks" of these oscillations in its predictions. Because of this, sin and cosine functions were applied to the hour, weekday, and week-of-year columns and added as additional features to the dataset. These sin/cos terms helped ensure that the cyclical nature of the hour/weekday/week-of-year features was encoded into the model. For example, these new features allowed our model to distinguish Week 52 of 2017 and Week 1 of 2018 as being only 1 week apart, rather than 51 weeks apart (as the model would see without the sin/cos terms accounting for this cyclic behavior).

The second improvement strategy was hyperparameter tuning search. Each test run of the hyperparameter tuning experiment altered one of the hyperparameters independently (while keeping each of the other hyperparameters constant) to evaluate that hyperparameter's effect on validation performance. Due to the time-series nature of our data, it would have been difficult to implement a method such as k-fold cross validation, thus each run was compared with the same validation set to keep the validation date range consistent. The varied hyperparameters are shown in the *Tested Values* column of Table III.

TABLE III
HYPERPARAMETER TUNING TESTED AND SELECTED VALUES

Hyperparameter	Tested Values	Selected Value
window size	[4, 12, 24, 168]	24
# layers	[1,2,3]	1
# hidden nodes	[10,25,50,75,100,150]	75
# training epochs	[1,3,5]	3
# activation func.	[tanh, sigmoid, relu, softmax]	tanh

Each of the hyperparameters varied in its effect on performance (i.e. validation set MAPE). The number of layers did not appear to provide any positive effect in performance (and also dramatically increased runtime), thus only 1 hidden layer was used in the final model. Moderate values of window size, number of hidden nodes per layer, and number of training epochs (i.e. number of passes through a given dataset) seemed to provide the best validation performance. Lower values of these hyperparameters tended to lead to high bias and an underfitting model, whereas higher values led to a high variance overfitting model (in addition to longer runtimes). Lastly, using a *tanh* activation function yielded the lowest MAPE, with *relu* and *sigmoid* both close seconds in performance and *softmax* much lower performance. After experimentation, it was found that the combination of hyperparameters shown in the *Selected Value* column of Table III yielded the lowest validation MAPE and was thus used for the final model.

The third improvement strategy was the creation of plots for validation and training error versus training set size, as described by Professor Ng in his lecture *Advice for Applying Machine Learning* [16]. First we produced this plot for a single city, Houston. The plot was difficult to interpret because the training and validation error trajectories were not smooth. We hypothesize that this erratic behavior came from an interaction between the way we set up our data folds for this experiment and the seasonality of our data. Similar to the outputs of *scikit-learn*'s *TimeSeriesSplit* function, our training and validation sets grew from earlier times to later times instead of being sampled randomly from our dataset [17]. Unfortunately, the strong seasonality of electricity demand, as seen in Figure 1, introduces substantive (not just random) variation in the composition of both training and validation datasets as they grow into new time periods. This may cause the erratic validation and training error trajectories.

Even though the error versus training set size plot for a single city was difficult to interpret, we noted that training error and validation error were still far apart even at large training sample sizes. We concluded that our model suffered from high variance. To combat this, we re-implemented our *python* code to allow it to process data from multiple cities at a time. After this change, we re-created the error versus training set size plot, this time using all cities' data at once (See Figure 3). The result was a plot which had a smaller gap between training and validation error. However, both errors were still higher than our target. This suggests that we had achieved lower variance, but still had high bias.

Fig. 3. Comparison of Training and Validation Error for Increasing Training Dataset Sizes

The final improvement strategy was experimenting with other neural network architectures known to perform well on time series data. These included stacking n layers of LSTMs, combining a CNN and an LSTM, and using a ConvLSTM which is specifically used for spatio-temporal modeling. We tried an implementation of each type of model and the MAPE of each model was compared to see which models would work best for our data. It was found that with the number of training examples in our dataset, single layer and stacked LSTMs (our default architectures) were the best options. This is because their MAPEs were less than 10%, whereas the ConvLSTM, CNN-LSTM and Time-Distributed layer networks produced MAPEs of around 18-22%.

TABLE IV
PERFORMANCE OF SELECT MODELS DESCRIBED BY MAPE AND MSE ON TEST DATASET (JANUARY 1, 2020 TO FEBRUARY 29, 2020)

Model		MAPE	MSE
Baseline	OLS regresssion	10.6%	2,083,500
	Autoregression	9.8%	1,966,059
LSTM	single city	4.6%	334,341
	multi-city	4.1%	111,985

Hahn, Meyer-Nieberg, and Pickl describe many short-term electricity demand forecasting models which predict a few hours in advance.. Most of these research or production models have a MAPE in the neighborhood of 2% [2]. Our models have a slight advantage over these models because we aim to predict on historical data. This means that we can include the weather variables for the exact hour we are predicting for. With this in mind, our goal was to build a model with a MAPE close to 2%. Table IV shows the errors of our models on the test dataset. Figure 4 shows predictions for three cities on the test set.

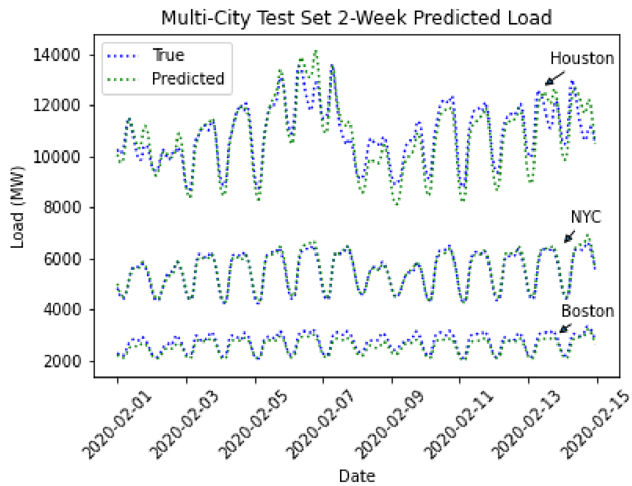

Fig. 4. Multi-City Test Set 2-Week Predicted Load

VI. Conclusion

A. Electricity Demand Predictions

We tried three different models to forecast the electricity demand across 5 cities in the US. Each progressively more refined model brought our MAPE error rate closer to the industry standard MAPE of 2% for similar models [2]. We found that the single-layer LSTM model performed the best, providing a MAPE of 4.1%. This model worked best because of its ability to model the non-linear relationship between weather, time and electricity demand.

For our next steps, we hope to collect more data across more cities and use it to create a spatio-temporal model using a combination of CNN, ConvLSTM and LSTM layers. This is to capture small seasonal patterns using the CNN and LSTM layers, and the ConvLSTM to capture the variability in patterns across cities.

B. Impact of COVID-19

With our trained LSTM model in hand, we could evaluate the impact of COVID-19 on electricity demand in U.S. cities. Figures 5 and 6 display the results for New York city, one of the first epicenters of the COVID-19 pandemic. A visible drop in peak electricity demand relative to our prediction is evident on Monday, March 16$^{\text{th}}$. ABC News' timeline of COVID-19 for New York City mentions that the first two confirmed COVID-19 deaths in New York City were reported on March 14$^{\text{th}}$, and that Governor Cuomo's first work-from-home order was put into effect on March 18$^{\text{th}}$. These events confirm the sudden drop in electricity demand seen in Figure 5.

Across all cities studied, the estimated changes in electricity demand due to COVID-19 are shown in Table V. These results are in the same range as similar results reported by Prol and O. [6]. While our model error rate on test data (MAPE: 4.1%) is large compared to many of these changes, we can be confident that the changes observed in Boston, Chicago, and New York City exceed the size of our error. In the future, a more thorough analysis of error by city could be undertaken to refine these conclusions.

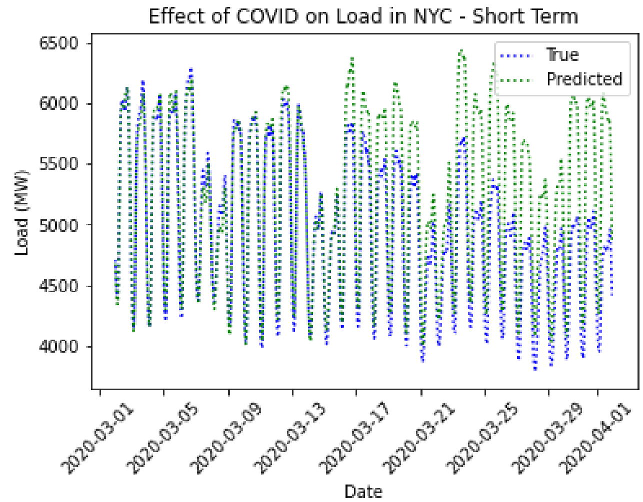

Fig. 5. Effect of COVID on Load in NYC - Short Term

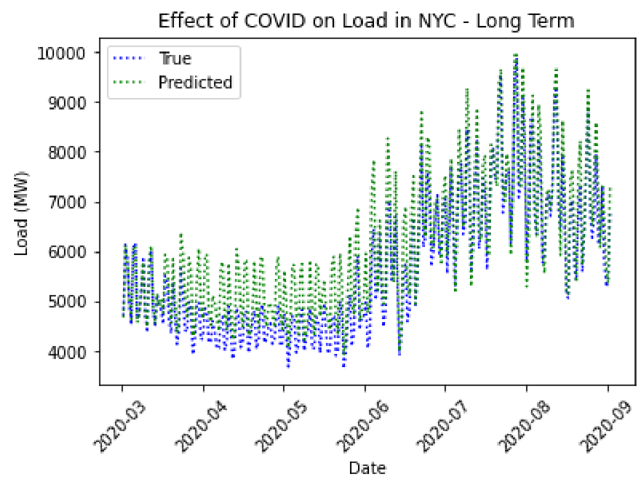

Fig. 6. Effect of COVID on Load in NYC - Long Term

TABLE V
Estimated Percentage Change in Total Electricity Demand Due to COVID-19 (March 15$^{\text{TH}}$ to June 30$^{\text{TH}}$, 2020)

City	Change in Total Demand
Boston	-4.8%
Chicago	-5.7%
Houston	+0.6%
Kansas City	-1.4%
New York City	-10.6%

VII. Code Repository

Our code for all models can be accessed at https://github.com/jgeiser47/CS229_Final_Project.

VIII. Contributions

A. Deep Dayaramani

I focused on brainstorming ideas on potential baseline models, data cleaning, experimental model analysis and producing the features for the data matrices for the 5 cities, exploring the second baseline model using Prophet and performing cross validation and visualization for the model. I worked on implementing the LSTM class combining Fletcher's work on scaling and model creation along with Josh's data visualization to make an easy interface for running experiments and creating visualizations. I also worked on implementations of the different model structures, to choose between single, stacked, ConvLSTMs and other models. Further implemented the test-on-splits function which provides the ability to create the train-validation error plots on different metrics.

B. Josh Geiser

I worked on researching potential topic ideas, performing experimental data analysis and plotting of the raw features, initial data parsing/cleaning into a more workable CSV format, setting up our GitHub repository structure, and training the Baseline 1 (linear regression) model. After the milestone, I implemented metric calculation capabilities to evaluate the performance of each of our trained models, added in the sin/cos features into our dataset, performed our hyperparameter tuning experiment, and worked on data visualization capabilities for the final report figures.

C. Fletcher Passow

I focused on understanding other studies assessing the impacts of COVID-19 on electricity demand. I also corresponded with our project TA about our question and what models would be most appropriate. I took the lead on writing the introduction and related work sections of this milestone report. Out of my discussions with our TA came our decision to use an LSTM neural network for our final model. After the milestone, I implemented the multi-city version of our LSTM code. I also took the lead in evaluating the impacts of COVID-19 based on the predictions that our model made.

References

[1] S. Fan and R. J. Hyndman, "Forecasting electricity demand in australian national electricity market," in *2012 IEEE Power and Energy Society General Meeting*, 2012, pp. 1–4.

[2] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making," *European Journal of Operational Research*, vol. 199, no. 3, pp. 902–907, Dec. 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221709002094

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735

[4] F. Gers, "Learning to forget: continual prediction with LSTM," in *9th International Conference on Artificial Neural Networks: ICANN '99*, vol. 1999. Edinburgh, UK: IEE, 1999, pp. 850–855. [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/cp_19991218

[5] A. Bahmanyar, A. Estebsari, and D. Ernst, "The impact of different COVID-19 containment measures on electricity consumption in Europe," *Energy Research & Social Science*, vol. 68, p. 101683, Oct. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214629620302589

[6] J. López Prol and S. O, "Impact of COVID-19 Measures on Short-Term Electricity Consumption in the Most Affected EU Countries and USA States," *iScience*, vol. 23, no. 10, p. 101639, Oct. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2589004220308312

[7] D. Agdas and P. Barooah, "Impact of the COVID-19 Pandemic on the U.S. Electricity Demand and Supply: An Early View From Data," *IEEE Access*, vol. 8, pp. 151 523–151 534, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9169615/

[8] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, 2nd ed. Melbourne, Australia: OTexts, 2018. [Online]. Available: OTexts.com/fpp2

[9] H. Hippert, D. Bunn, and R. Souza, "Large neural networks for electricity load forecasting: Are they overfitted?" *International Journal of Forecasting*, vol. 21, no. 3, pp. 425–434, Jul. 2005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016920700400130X

[10] G. Ruan, D. Wu, X. Zheng, H. Zhong, C. Kang, M. A. Dahleh, S. Sivaranjani, and L. Xie, "A Cross-Domain Approach to Analyzing the Short-Run Impact of COVID-19 on the U.S. Electricity Sector," *arXiv:2005.06631 [cs, eess, math]*, Aug. 2020, arXiv: 2005.06631. [Online]. Available: http://arxiv.org/abs/2005.06631

[11] S. J. Taylor and B. Letham, "Forecasting at scale," PeerJ Preprints, preprint, Sep. 2017. [Online]. Available: https://peerj.com/preprints/3190v2

[12] J. J. Q. Yu, A. Y. S. Lam, D. J. Hill, and V. O. K. Li, "Delay aware intelligent transient stability assessment system," *IEEE Access*, vol. 5, p. 17230–17239, 2017.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, Nov. 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[15] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[16] A. Ng, "Advice for applying Machine Learning," Stanford University, Nov. 2020. [Online]. Available: http://cs229.stanford.edu/materials/ML-advice.pdf

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.