INL/RPT-24-04401
Revision 0

Optimizing Prediction Error for Time-dependent Solar Radiation Modeling

March 2024

Dylan McDowell and Aaron Epiney
Idaho National Laboratory

Robert Flanagan and Mark Deinert
Colorado School of Mines

IES
Integrated Energy Systems

INL/RPT-24-04401
Revision 0

Optimizing Prediction Error for Time-dependent Solar Radiation Modeling

Dylan McDowell and Aaron Epiney
Idaho National Laboratory
Robert Flanagan and Mark Deinert
Colorado School of Mines

March 2024

Idaho National Laboratory
Integrated Energy Systems
Idaho Falls, Idaho 83415

http://www.ies.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Science
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Page intentionally left blank

FOREWORD

Submission of the attached report formally fulfills the completion of the Integrated Energy Systems (IES) Level 3 Milestone (M3CT-24IN1206032), "Demonstrate algorithms for training and generating stochastic synthetic time series," due March 31, 2024.

Page intentionally left blank

CONTENTS

FIGURES

Page intentionally left blank

ACRONYMS

ANN	artificial neural network
ARMA	autoregressive moving-average
DHI	diffused horizontal irradiance
LSTM	long short term memory
ML	machine learning
NN	neural network
NREL	National Renewable Energy Laboratory
NSRDB	National Solar Radiation Database

Page intentionally left blank

Optimizing Prediction Error for Time-dependent Solar Radiation Modeling

1. INTRODUCTION

The variability of renewable energy generators is expected to place increasing stress on the existing electrical grids in the United States [1]. Thus, accurate forecasting of renewable energy generation is essential for alerting grid operators as to how much compensatory generation is needed [2,3], and for enabling efficient bidding in electricity markets. Renewable energy providers will also want this information so they can know how much electricity they can provide. Historically, statistical methods have been used to predict solar radiation. Techniques such as autoregressive moving-average (ARMA) and autoregressive integrated moving-average have shown some promise in terms of such forecasting [4–7]. The "inertia" in these models makes them ideal for short-term predictions, but their accuracy drops off as the time horizon of the prediction increases.

Machine learning (ML) represents another approach to predicting solar signals. Kassa et al. (2016) used an artificial neural network (ANN) to predict wind power, achieving success at predicting short-term (hour-ahead) wind generation on a microgrid [8]. Liu et al. (2019) used a discrete wavelet transform combined with an long short term memory (LSTM) network, demonstrating better performance than that afforded by wavelet combinations with regression neural networks (NNs) and backpropagation NNs [9]. Delgado et al. (2020) showed that an LSTM will outperform both a moving average statistical approach as well as a multilayer perceptron model in terms of mean square error results [10].

Similar work has also been applied to solar energy [11,12]. Demolli (2019) used a collection of ML techniques (e.g., Lasso regression, k-nearest neighbors regression, and XGBoost regression) to make both near- and long-term wind power predictions [13]. Tang (2018) also utilized a Lasso-based approach to predict solar power generation, using 5-minute weather data from Massachusetts [14]. It has been argued that a variety of methods should be used, as several reviews of the space have shown that ML method performance depends on geographical area [15–17]. Other reviewers have combined statistical techniques with ML. Gomes (2012) achieved promising results by combining ARMA with an ANN in order to predict wind speed [18]. Xie (2018) combined variational mode decomposition, deep belief networks, and ARMA in order to predict solar signals, achieving better results than could be obtained from using each separate method individually [19]. LSTM NNs, when applied to hour-ahead solar signal predictions, outperformed a backpropagation NN, an ANN, and the least squares regression method [20,21]. Jeon et al. (2020) demonstrated that LSTM can be used to make hour-ahead solar signal predictions based on the conditions observed in the current hour as well as in the previous 24. For the city of Incheon in South Korea, they achieved an hourly root mean square error of 17.4 W/m^2 by comparing the historical solar signals from the first half of the year against the predicted signals for that same period. They also made predictions based on other cities around the globe (Denver, Cape Town, Paris), achieving an hourly root mean square error of 30.1 W/m^2 over a 6 month period [22]. Rajagukguk et al. (2020) reviewed deep learning models for solar forecasting, finding that hybrid models (e.g., convolution-LSTM) outperformed standalone ones (e.g. simply LSTM). However, the studies reviewed were for a wide variety of locations, and the best modeling results pertained to Alice Springs, Australia, which is a notably dry area.

Here, we present an approach for optimizing an LSTM model for solar predictions, and we use that model to calculate the error associated with those predictions. We demonstrate this approach in the context of two U.S. locations: Austin, TX, and Seattle, WA. The predictions are based on historical weather condition data from the National Renewable Energy Laboratory (NREL) National Solar Radiation Database (NSRDB) [23]. Data on the two cities and their surrounding areas were relied upon for making accurate predictions. The LSTM methodology and data preparation method are covered in the following section.

2. METHODS

The synthetic results were generated using a LSTM NN. Training data on Austin, TX, and Seattle, WA, were collected from the NREL NSRDB [23] and the U.S. Department of Energy's Open Energy Data Initiative, and included wind speed, wind direction, solar strength, dew point, temperature, humidity, air pressure, time of day, and time of year data. Removing the historical value from the calculated solar maximum opened an additional field in which to represent cloud cover. Historical data were used to predict a year's worth of data, and the prediction results were compared against the historical data via an R^2 measure.

Data Processing. The data used in this work came from the NREL NSRDB 23]. Eight years' worth of data on Austin, TX, and Seattle, WA, were acquired, along with data on 24 different areas surrounding those cities. Eight of those areas directly abutted one city or the other, and the remaining 16 were located on the outskirts of the initial 8. The weather condition data from these additional regions—whether up- or downstream of the target area—served to provide pseudo-forecast data. The data used for this work covered the following: wind speed, wind direction, solar strength, dew point, temperature, humidity, air pressure, time of day, and time of year. All the data fields, apart from diffused horizontal irradiance (DHI), were scaled from zero to one in order to improve NN performance.

Two additional data fields were generated from the raw data: normalized DHI and estimated cloud cover. Normalized DHI represents the value obtained after normalizing the DHI against the maximum solar irradiance at the same time. This was used as a prediction field. The estimated cloud cover was determined by subtracting the historical DHI signals from the maximum solar irradiance, normalized against the maximum solar irradiance. This value represents the percentage of sunlight blocked due to clouds.

The data were presented in 30-minute increments. The solar signal was shifted forward 1 and 24 hours, such that conditions from previous time steps could be used to predict the shifted solar and wind signals. Six years' worth of data were used to train the NN, and an additional 2 years' worth were used to test it. This split was made to ensure that the NN could catch differences from year to year. It is possible that, were only a single year used for testing, the NN might fit that year well but not another (due to large yearly climate impacts such as localized droughts).

Synthetic solar data were updated to incorporate the maximum solar irradiance as calculated at any time by replacing values with zeroes anytime the maximum solar irradiance is also zero. This is acceptable because the maximum solar irradiance can be calculated in advance and does not need to be predicted using the ML algorithm.

NN Setup. LSTM NNs analyze sequences of data in a time series—unlike feedforward NNs, which process single data points. This is ideal for time series data because it allows for "memory" to impact predictions of future data. The structure of an LSTM is commonly described as a cell containing three "layers" that manage the information in that cell: input layer gates decide which data are updated within the cell, forget layer gates determine which data are maintained and which are dropped, and output layer gates return the result of the cell [24,25]. For this work, the LSTM NNs were built using the Keras ML package provided by TensorFlow [26], itself a Python ML package. A separate ML network was set up for each signal and prediction length.

3. PREDICTION METHODS AND RESULTS

3.1 Gaussian Prediction Method

This method utilizes the aforementioned historical data, and optimizes the LSTM by applying a loss function that fits the residuals to a Gaussian distribution by varying the mean and standard deviation. Thus, this method generates both a predicted value and a standard deviation (variance) associated with that value. The workflow for this method and the results obtained by using it are shown in Figure 1 and Figure 2, respectively.

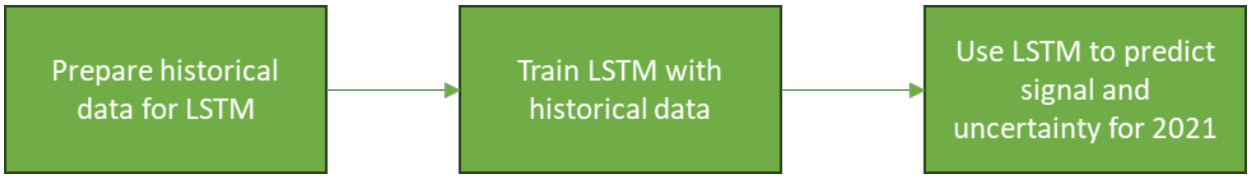

Figure 1. Flow chart of the Gaussian prediction method.

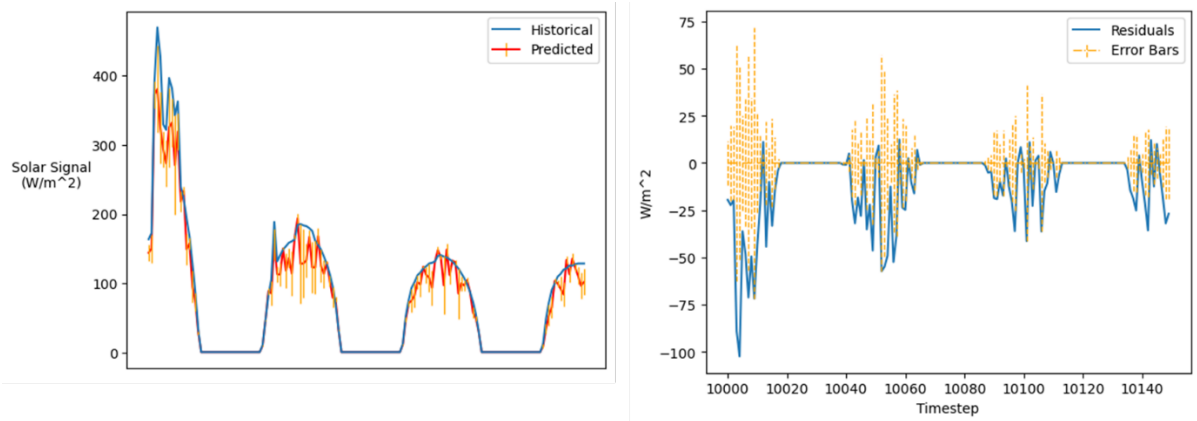

Figure 2. Plots of predicted vs. historical (left) and residual (right) data, using the Gaussian prediction method.

Unfortunately, the Gaussian prediction method requires that the values being predicted fit a normal distribution—something that does not hold true for all time periods of a given day. For example, Figure 3 shows histograms of the solar signal for the noon hour in both January and June for Austin, TX. Therefore, additional methods were attempted. A comparison of all the different methods is given at the end of the document. The same timeframe was applied to each method to enable equal comparison between all the methods presented in this work.

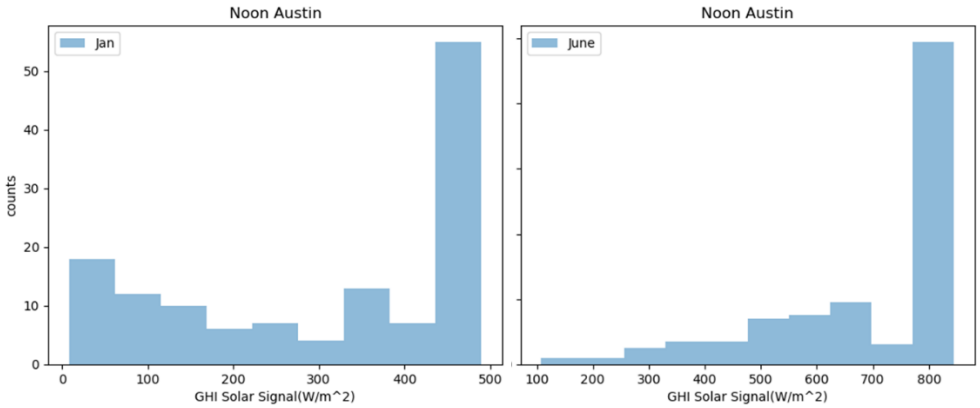

Figure 3. Histograms of the solar signal for the noon hour in both January (left) and June (right) for Austin, TX.

3.2 LSTM Informed by Error

In this method, the LSTM was utilized to predict the solar signal, then the residuals of the results were used to calculate a standard deviation for each time step. Next, a second LSTM was trained on the historical weather data and standard deviation for each time step. This new LSTM then predicted a new standard deviation for each time step. The workflow for this method and the results obtained by using it are shown in Figure 4 and Figure 5, respectively.

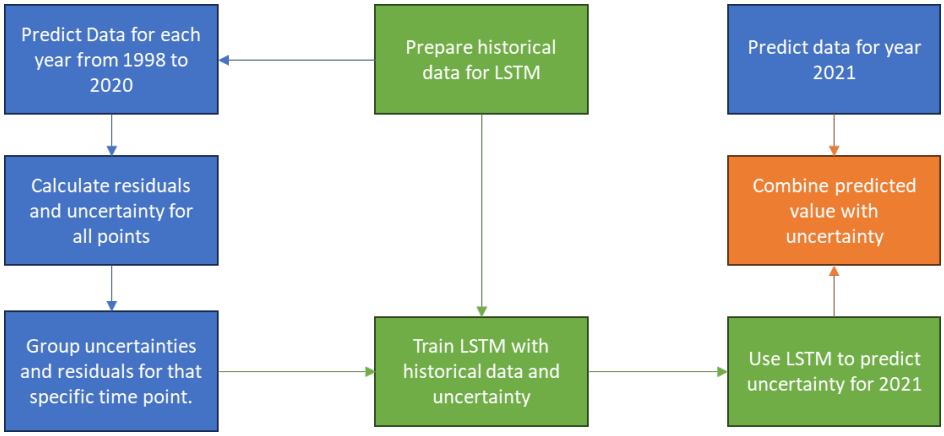

Figure 4. Flow chart of the LSTM-informed-by-error method.

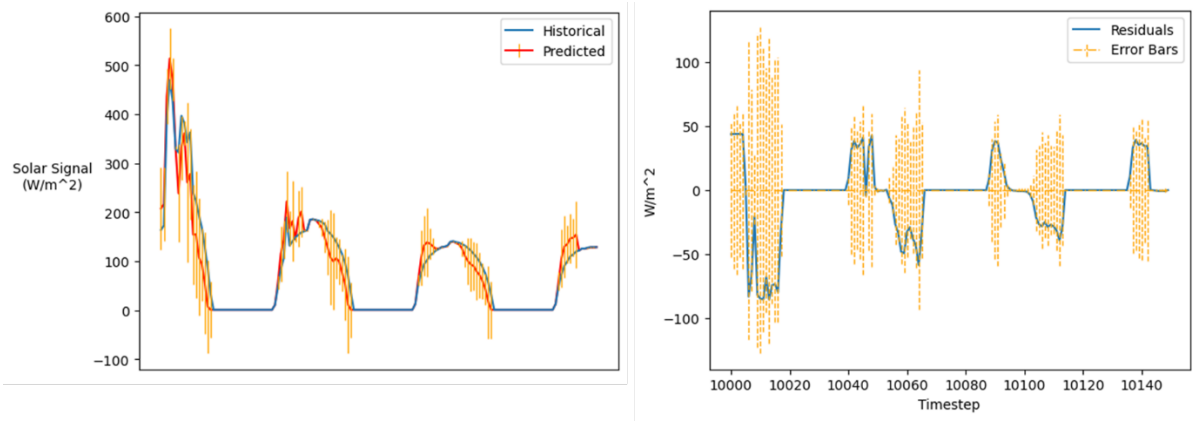

Figure 5. Plots of predicted vs. historical (left) and residual (right) data, using the LSTM-informed-by-error method.

The results here indicate a better fit of the predicted values, but larger error bars are associated with the predictions.

3.3 Gompertz Distribution Inputs

The next investigated method utilized the Gompertz distribution to inform the LSTM prediction. This was accomplished by grouping like time steps from the days surrounding. An example of this was to group the solar signals for 7am from January 7 through January 15 of each year. These values were then used to generate a cumulative histogram to which a Gompertz distribution was then fit. The parameters of that fitted distribution were used as inputs to the LSTM, which was then used to predict the three Gompertz parameters for each distribution. From those predicted parameters, a standard deviation and mean was calculated. The workflow of this process and the results obtained by using it are given in Figure 6 and Figure 7, respectively.

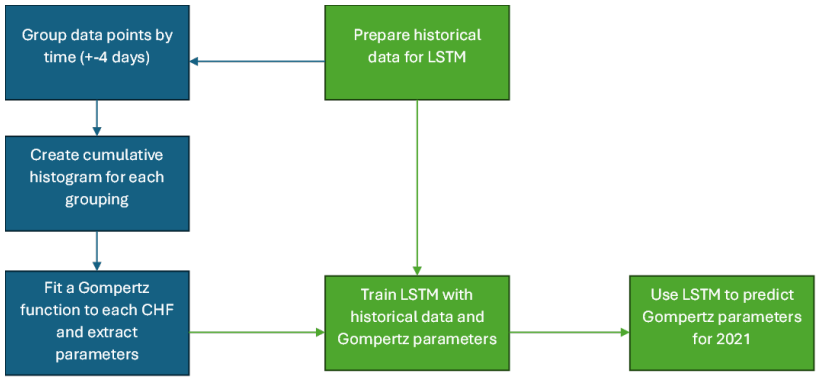

Figure 6. Flow chart of the Gompertz distribution inputs method.

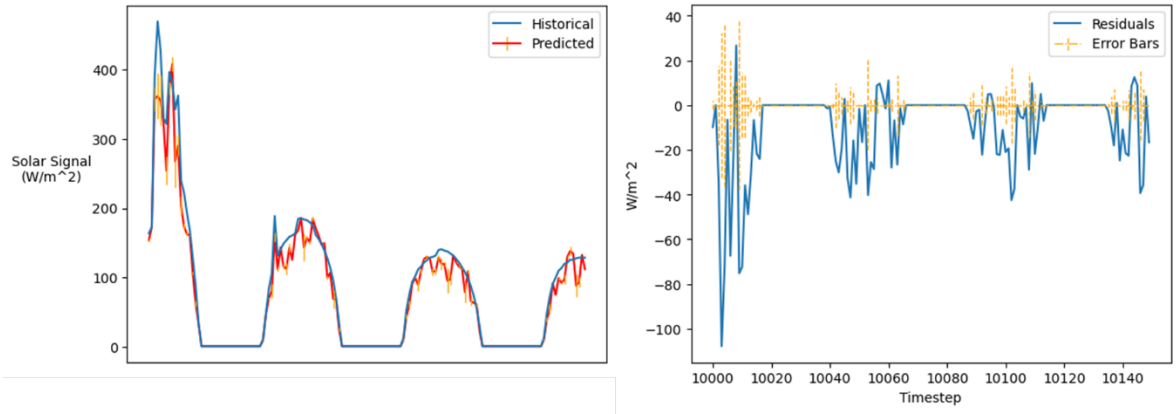

Figure 7. Plots of predicted vs. historical (left) and residual (right) data, using the Gompertz distribution inputs method.

In general, these results show much smaller error and residuals. However, the residuals exceed the error in a significant number of time steps, and error magnitude peaks are seen where the solar signal is largest.

3.3.1 Gompertz Prediction of Error

Another method involving the Gompertz distribution utilized LSTM to predict the solar signal from the historical weather data and the fitted Gompertz parameters from the first Gompertz method. At the same time, another LSTM was used to predict a variance. This was accomplished by converting the Gompertz parameters into a variance. Then those variances were used as training inputs to the LSTM. Finally, the variance was predicted using the second LSTM, and the results were then grouped. The workflow and results achieved by using this method are given in Figure 8 and Figure 9 respectively.

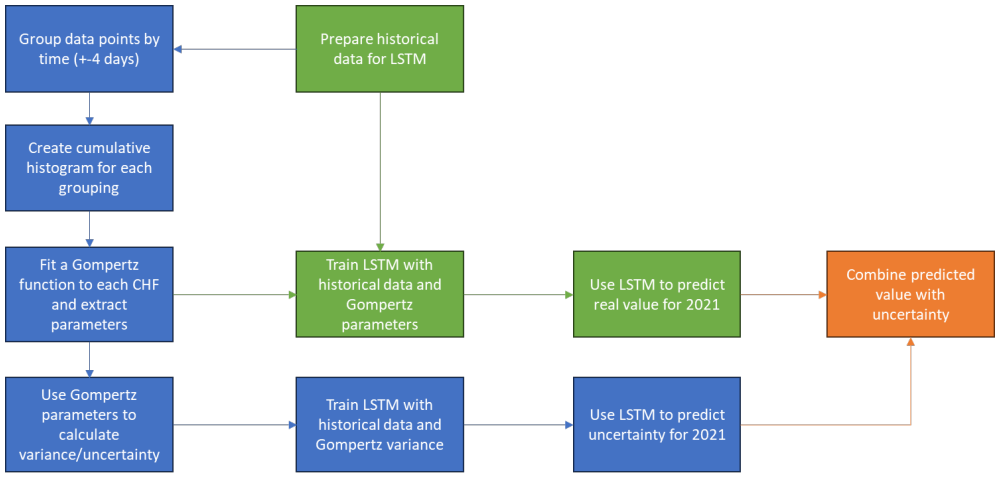

Figure 8. Flow chart of the Gompertz-prediction-of-error method.

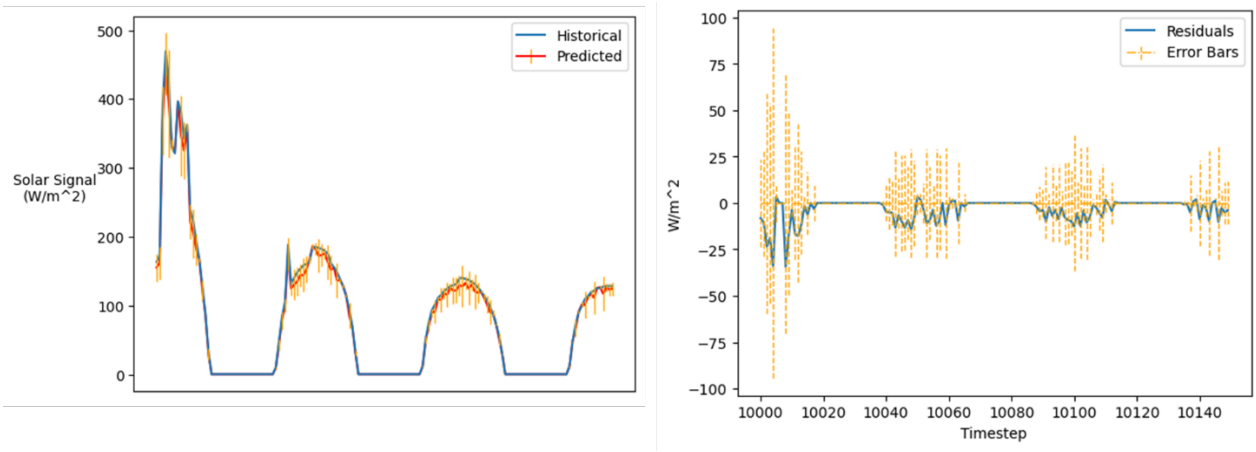

Figure 9. Plots of predicted vs. historical (left) and residual (right) data, using the Gompertz-prediction-of-error method.

These results show predictions that are an improvement over those of the previous methods (smaller residuals), but reflect larger error bars than seen with the previously discussed Gompertz.

3.3.2 Alternative Gompertz Prediction of Error

In this method, an LSTM trained on the Gompertz parameters was employed to predict a solar signal that was itself used to calculate the residuals for each time step. The residuals were then grouped (as in previous steps) to generate a Gompertz distribution for each time step, based on the residuals. That Gompertz distribution of the residuals was then used to calculate a variance. Next, the parameters and variance were used to train a new LSTM for predicting a variance. The results were combined. The workflow and results obtained by using this method are given in Figure 10 and Figure 11, respectively.

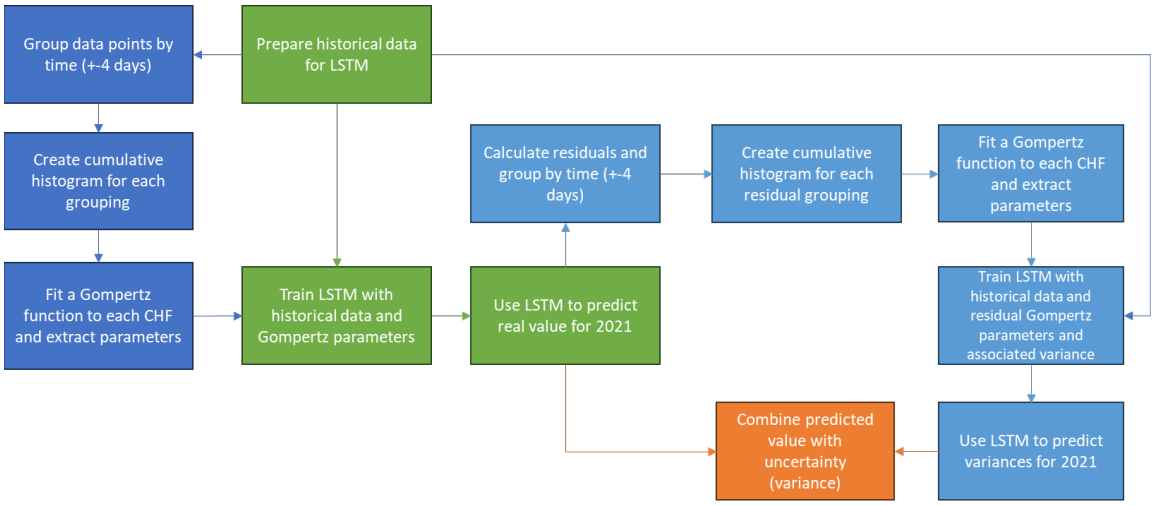

Figure 10. Flow chart of the alternative Gompertz-prediction-of-error method.

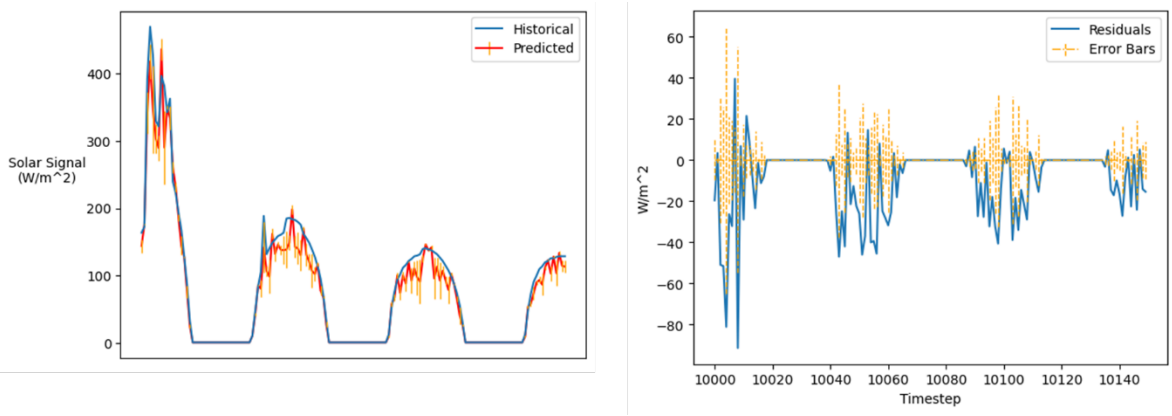

Figure 11. Plots of predicted vs. historical (left) and residual (right) data, using the alternative Gompertz prediction-of-error method.

With this method, the residual values are larger than seen from the previous method, but the errors are significantly smaller.

3.3.3 LSTM with Gompertz Prediction of Error

The final method investigated in this work simplifies the previous process. It predicts the solar signal by using only the historical weather data, but also applies the error calculation approach from the previous method. The workflow and the results obtained by using this method are given in Figure 12 and Figure 13, respectively.

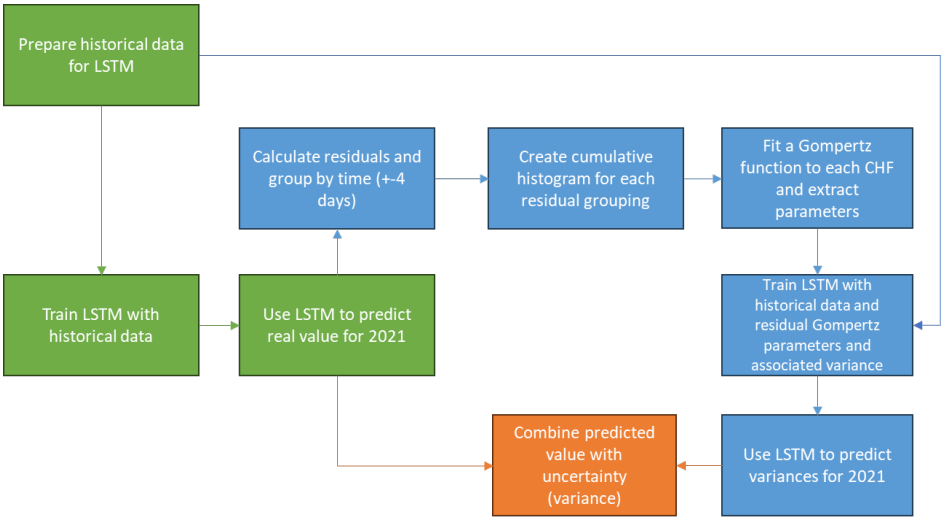

Figure 12. Flow chart of the LSTM-with-Gompertz-prediction-of-error method.

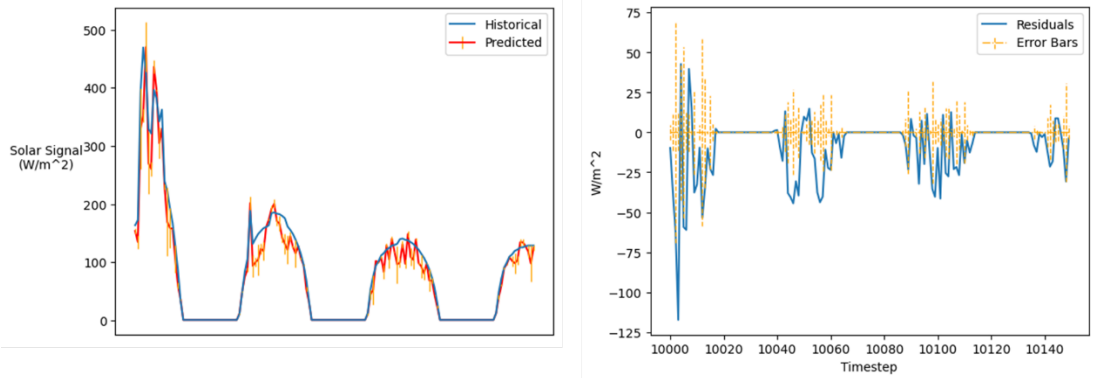

Figure 13. Plots of the predicted vs. historical (left) and residual (right) data, using the LSTM-with-Gompertz-prediction-of-error method.

This method produces residuals of the same size as those seen in the previous method (within 10%), but with smaller error bars. However, neither this nor the previous method match the performance of the Gompertz-prediction-of-error method.

4. CONCLUSIONS

These six methods represent different ways of predicting solar signals, and all six carry different degrees of associated error. The novel component of this work is the predictions of the error bars associated with each predicted value, instilling a measure of confidence in the results obtained from each method. Per the figures provided here (and the data they convey), the best results were obtained via the Gompertz-prediction-of-error method. Despite the fact that, for each prediction, its error bars greatly exceeded the residuals, the error bars were nonetheless smaller than the residuals from all the other methods. Further analysis of these results will be forthcoming in a future journal publication.

5. REFERENCES

1. Anees, A. S. 2012. “Grid Integration of Renewable Energy Sources: Challenges, Issues and Possible Solutions.” In proceedings of the 2012 IEEE 5th India International Conference on Power Electronics (IICPE), Delhi, India, December 2012.
2. Ciarreta, A., C. Pizarro-Irizar, and A. Zarraga. 2020. “Renewable Energy Regulation and Structural Breaks: An Empirical Analysis of Spanish Electricity Price Volatility.” *Energy Economics* 88: 04749. https://doi.org/10.1016/j.eneco.2020.104749.
3. Schmietendorf, K., J. Peinke, and O. Kamps. 2017. “The Impact of Turbulent Renewable Energy Production on Power Grid Stability and Quality.” *European Physical Journal* 90: 222. doi:10.1140/epjb/e2017-80352-8.
4. Huang, R., T. Huang, R. Gadh, and N. Li. 2012. “Solar Generation Prediction Using the ARMA Model in a Laboratory-Level Micro-Grid.” In proceedings of the 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm), Tainan, Taiwan, November 2012.
5. Wang, J., Q. Zhou, and X. Zhang. 2018. “Wind Power Forecasting Based on Time Series ARMA Model.” *IOP Conference Series: Earth and Enviornmental Science* 199: 022015. doi:10.1088/1755-1315/199/2/022015.
6. Zhang, J. and C. Wang. 2013. “Application of ARMA Model in Ultra-Short Term Prediction of Wind Power.” In proceedings of the 2013 International Conference on Computer Sciences and Applications; Wuhan, China, December 2013.

7. Colak, I. et al. 2015. "Multi-Period Prediction of Solar Radiation Using ARMA and ARIMA Models." In proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, December 2015.
8. Kassa, Y. et al. 2016. "A GA-BP Hybrid Algorithm Based ANN Model for Wind Power Prediction." In proceedings of the 2016 IEEE Smart Energy Grid Engineering (SEGE); Oshawa, ON, Canada, August 2016.
9. Liu, Y. et al. 2019. "Wind Power Short-Term Prediction Based on LSTM and Discrete Wavelet Transform." *Applied Sciences* 9(6): 1108. doi:10.3390/app9061108.
10. Delgado, I. and M. Fahim. 2020. "Wind Turbine Data Analysis and LSTM-Based Prediction in SCADA System." *Energies* 14(1): 125. https://doi.org/10.3390/en14010125.
11. İzgi, E. et al. 2012. "Short–Mid-Term Solar Power Prediction by Using Artificial Neural Networks." *Solar Energy* 86(2): 725–733. doi:10.1016/j.solener.2011.11.013.
12. Abuella, M. and B. Chowdhury. 2015. "Solar Power Forecasting Using Artificial Neural Networks." In proceedings of the 2015 North American Power Symposium (NAPS), Charlotte, NC, USA, October 2015.
13. Demolli, H. et al. 2019. "Wind Power Forecasting Based on Daily Wind Speed Data Using Machine Learning Algorithms." *Energy Conversion and Management* 198: 111823. doi:10.1016/j.enconman.2019.111823.
14. Tang, N. et al. 2018. "Solar Power Generation Forecasting With a LASSO-Based Approach." *IEEE Internet Things Journal* 5(2): 1090–1099. doi:10.1109/JIOT.2018.2812155.
15. Voyant, C. et al. 2017. "Machine Learning Methods for Solar Radiation Forecasting: A Review." *Renewable Energy* 105: 569–582. doi:10.1016/j.renene.2016.12.095.
16. Guermoui, M. et al. 2020. "A Comprehensive Review of Hybrid Models for Solar Radiation Forecasting." *Journal of Cleaner Production* 258: 120357. doi:10.1016/j.jclepro.2020.120357.
17. Kumari, P. and D. Toshniwal. 2021. "Deep Learning Models for Solar Irradiance Forecasting: A Comprehensive Review." *Journal of Cleaner Production* 318: 128566. doi:10.1016/j.jclepro.2021.128566.
18. Gomes, P. and R. Castro. 2012. "Wind Speed and Wind Power Forecasting Using Statistical Models: AutoRegressive Moving Average (ARMA) and Artificial Neural Networks (ANN)." *International Journal of Sustainable Energy Development* 1: 41–50. doi:10.20533/ijsed.2046.3707.2012.0007.
19. Xie, T. et al. 2018. "A Hybrid Forecasting Method for Solar Output Power Based on Variational Mode Decomposition, Deep Belief Networks and Auto-Regressive Moving Average." *Applied Sciences* 8(10): 1901. doi:10.3390/app8101901.
20. Huang, X. et al. 2020. "A Comparison of Hour-Ahead Solar Irradiance Forecasting Models Based on LSTM Network." *Mathematical Problems in Engineering* 2020: 1–15. doi:10.1155/2020/4251517.
21. Qing, X. and Y. Niu. 2018. "Hourly Day-Ahead Solar Irradiance Prediction Using Weather Forecasts by LSTM." *Energy* 148: 461–468. doi:10.1016/j.energy.2018.01.177.
22. Jeon, B. and E.-J. Kim. 2020. "Next-Day Prediction of Hourly Solar Irradiance Using Local Weather Forecasts and LSTM Trained with Non-Local Data." *Energies* 13(20): 5258. doi:10.3390/en13205258.
23. Sengupta, M. 2018. "The National Solar Radiation Data Base (NSRDB)." *Renewable and Sustainable Energy Reviews* 89: 51–60.

24. Hochreiter, S. and J. Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9: 1735–1780. doi:10.1162/neco.1997.9.8.1735.

25. Staudemeyer, R. C. and E. R. Morris. 2019. “Understanding LSTM -- a Tutorial into Long Short-Term Memory Recurrent Neural Networks.” Cornell University. doi:10.48550/ARXIV.1909.09586.

26. Abadi, M. et al. 2016. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.” Cornell University. https://doi.org/10.48550/arXiv.1603.04467.

27. Nix, D. A. and A. S. Weigend. 1994. “Estimating the Mean and Variance of the Target Probability Distribution.” In proceedings of the Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94), Orlando, FL, USA, June 28–July 02, 1994. https://doi.org/10.1109/ICNN.1994.374138.

28. Wilkinson, I. M. et al. 2022. “Confidence Estimation in the Prediction of Epithermal Neutron Resonance Self-Shielding Factors in Irradiation Samples Using an Ensemble Neural Network.” *Energy and AI* 7: 100131. https://doi.org/10.1016/j.egyai.2021.100131.