**Optimal Forecasting of Residential Energy Sales Through SARIMA Aware LSTM Networks and Initial Seed Optimization**

Ethan Norton  
Northwestern University  
https://github.com/EthanNorton/MSDS-alignment/tree/MSDS-458  
EthanRNorton@protonmail.com  
July 27, 2025

Table of Contents

[Abstract 3](#_Toc204524802)

[Introduction and Problem Statement 4](#_Toc204524803)

[Background and Literature Review 5](#_Toc204524804)

[Data 6](#_Toc204524805)

[Research Design and Modeling Method(s) 7](#_Toc204524806)

[Design and Implementation Considerations 9](#_Toc204524807)

[Results 9](#_Toc204524808)

[Analysis and Interpretations 13](#_Toc204524809)

[Conclusions 14](#_Toc204524810)

[Future Direction 15](#_Toc204524811)

[Bibliography 15](#_Toc204524812)

# Abstract

The purpose of this study is to design a novel custom energy consumption prediction model based on LSTM Deep-Learning methodology regarding sequential learning, utilizing data from the U.S. Energy Information Administration (EIA) residential sector. This project leverages a Long Short-Term Memory (LSTM) neural network enhanced with a SARIMA-informed seasonal weighting feature to forecast energy sales, in combination with a Fourier transformed month variable to enable seasonality detection. A seed sensitivity analysis across 50 initializations was conducted to explore the impact of model randomness on performance. Additionally, combinatorics of hyperparameter tuning in conjunction with random seeds were evaluated to engineer metrics such as R², MAE, RMSE, and MAPE. These metrics were evaluated to determine the effectiveness and reproducibility of results. Then, the average of 30 seed initializations was taken to determine if LSTM or SARIMA informed LSTM was a better model for the problem.

**Keywords**: Seasonal Forecasting; Deep-Learning; LSTM; energy consumption pattern; SARIMA; Fourier Time Series Analysis

# Introduction and Problem Statement

The acceleration of large language models (LLMs) and AI workloads has placed demands on energy infrastructure, particularly from data centers and GPU-intensive computational environments. As AI adoption continues to scale and agents continue to be implemented, ensuring energy system stability, efficiency, and sustainability becomes a complex problem. Electricity consumption forecasting plays a crucial role in this context, demanding meticulous energy planning, distribution optimization, and infrastructure development. Accurate forecasting supports utilities and governments in enhancing grid reliability, managing peak loads, and anticipating demand fluctuations. Traditional statistical models such as ARIMA and linear regression often fall short in capturing the non-linear, temporal dynamics of energy usage. Deep learning models, such as Recurrent Neural Networks (RNNs) and their advanced variant, Long Short-Term Memory (LSTM) networks, which were introduced by (Hochreiter and Schmidhuber, 1997), have emerged as powerful tools for modeling these complex sequential relationships. However, LSTMs in the context of this problem, were not fully capturing the impact of seasonality, even with summer month feature engineering.

Therefore, in this study, LSTM networks and SARIMA aware LSTM are implemented along with precise feature engineering techniques to predict energy consumption. Engineered variables such as revenue per customer, year-over-year consumption growth, and cyclical monthly patterns are incorporated to optimize model fit and improve interpretability. The integration of AI techniques for energy forecasting aligns with the broader need for intelligent, resource-efficient infrastructure in an era increasingly defined by AI. It is believed that applying deep learning to the energy landscape will contribute greatly to the development of further advancements in AI, creating a positive feedback loop.

# Background and Literature Review

As an introduction to gain background on the material, this paper was explored to evaluate the process of training the LSTM deep learning model, which is a subset of the RNN model (Hrnjica and Mehr, 2019). However, this methodology could be complex, and in order to provide a more accurate and robust model, the best solution in this scenario is to combine it with modern deep learning theory and integrating other time series analysis principles. The LSTM’s forecasting mechanism has been widely used for many time-series forecasting in recent years (Choi, Cho, and Kim, 2020).

One study (Emshagin, Halim, and Kashef, 2022) deeply laid out the architecture of the LSTM, which was fundamental in applying the algorithm to the EIA dataset. The LSTM network has three main gates; input gate, output gate and forget gate. Input gates remember important and past steps, while forget gates delete nonpattern related or significant data. Features, such as the ones engineered in this study, will be passed through the LSTM network during the training phase.

The methodology of this paper incorporates a sliding window mechanism. This was influenced and refined by the application of another paper (Kaur et al., 2020). The sliding window method is used to transform time series data, in our case months, to a regression problem. In our example, the window was set to 12 to capture one per month, with applying a summer weight and a premium weight appended to the sequential mapping. Similarly to (Kaur et al., 2020), this paper also applies MinMaxScaler.

In order to shape the parameter tuning and hyperparameter optimization, literature was reviewed in order to determine a baseline of what has been performed in similar literature. One study, (Choi, Cho, and Kim, 2020) explored 40 different settings in the hyperparameter setting. Tanh and stochastic gradient descent were explored, in order to determine which would be a better fit for the model. This was directly applicable to the methodology presented in this paper, however a ReLu activation function was chosen due to performance.

The applications and importance of Fourier transformed LSTMs are grounded in modern literature (Ren, Xu, and Duan, 2022). Their paper on LSTM stock prediction under oil shocks likely had more volatility and perhaps less seasonality. However, it emphasizes the importance and relevance of performing Fourier transformations on time series data that is monthly/cyclical.

One paper, (Roy, Ishmam, and Taher, 2021) evaluated the application of many deep learning methodologies – both deterministic and probabilistic, and emphasized the potential for hybrid models “Statistical methods face challenges from their inability to process big data, while AI-based methods have other shortcomings in terms of model complexity and dependence on large training datasets”. As summarized, hybrid approaches involve two more methods to blend into a holistic approach.

The motivation for evaluating various dropout rates was inspired by (Srivastava et al., 2014), these parameters were finalized by performing hyperparameter tuning combinations and determining which fit the model the best.

# Data

Monthly data from the U.S. Energy Information Administration (EIA) was used, covering electricity sales, revenue, and customer counts, which was downloaded via API. Data was limited to a specific state-level granularity from January 2001 to December 2023, with 5 states (California, Texas, New York, Florida, Illinois), and 3 sectors (Residential, Commercial, Industrial) displaying 276 total observations.

**Figure 1:** Data set details, where the data was downloaded via API from the EIA.

A screen shot of a computer

AI-generated content may be incorrect.

**Key derived engineered features include:**

* **Average consumption per customer**: Total energy sales divided by the number of customers, reflecting household usage trends.
* **Revenue per customer**: Revenue divided by customer count, capturing economic behavior related to electricity use.
* **Year-over-year consumption growth**: The 12-month percentage change in average consumption, highlighting long-term demand shifts.
* **Customer growth rate**: The month-over-month percent change in the number of customers, which captures grid expansion dynamics.
* **Price elasticity**: Estimated by dividing the percent change in average consumption by the percent change in price, used to measure demand sensitivity.
* **Sine and cosine month encodings**: Used to represent cyclical patterns in electricity usage across the calendar year, with a fourier improved transform as well.
* **Binary peak-season indicator**: A binary variable indicating summer months (June through September) when demand typically spikes.
* **SARIMA-informed sigmoid weighting**: A smooth transformation of YoY growth to capture SARIMA-derived seasonality for weighting.
  + Instead of training a SARIMA model in combination or on its own, this study is built upon the core principles of SARIMA. Specifically, its weight on capturing seasonal fluctuations and periodic demand shifts, to inform and boost LSTM training weights. A smoothed sigmoid transformation of year-over-year consumption growth was applied to generate a continuous seasonal signal. This SARIMA-informed signal was next architected to generate a seasonal weighting multiplier, which slightly boosted or suppressed training importance based on consumption volatility and time of year. Additionally, a separate summer weighting factor was derived based on the average energy consumption differential between summer and non-summer months. Together, these adjustments introduced *domain-informed seasonal regularization* into the LSTM training pipeline, structuring the model to learn seasonally aware patterns while avoiding hard-coding explicit lags or differencing – in order to avoid overfitting.
* **Summer premium multiplier**: An adjustment applied to summer months based on relative consumption intensity compared to non-summer periods.
  + **Borrowed from previous week on Kaggle housing dataset of a successful feature of cubing the square footage, this was also cubed to emphasize the importance of summer.**
* **Final sample weighting**: A combination of SARIMA-informed weights and the summer premium used during training to emphasize seasonally important periods.

All continuous variables were normalized using StandardScaler. No dimensionality reduction (e.g., PCA) was applied.

# Research Design and Modeling Method(s)

This project uses a univariate and multivariate LSTM modeling approach implemented in TensorFlow/Keras. The model architecture includes:  
- One or more LSTM layers, which ranged for 32 to 128 layers.  
- Dropout regularization, which ranged from .2 to .3.   
- ReLU, sigmoid and tanh activation experiments  
- Dense linear output layer, ranging from 16 to 32.

**Figure 2:** An example of loading parameters into the model that will be used to train the model, and output various R2s.

A black screen with text

AI-generated content may be incorrect.  
Data is reshaped into overlapping 3D sequences of the form (samples, time steps, features). Models were trained using the Adam optimizer and MSE loss. A `ReduceLROnPlateau` learning rate scheduler and early stopping were used for training control.  
Several architectural variants were tested to observe the impact of activation function, window size, and dropout on model performance. The design logic focused on balancing capacity and regularization to avoid overfitting on limited training samples. The data was also split into a 80/20 train test ratio.

Two various tables performed various analyses on hyperparameters to analyze. ReLu was deemed to be the better activation function, and the parameters that were utilized to train the final model were a learning rate of .0005, two LSTM layers of 64 each, dense layers of 16, dropout rates of .2 and .3.

**Table 1: R2 performance of various sigmoid activation functions, learning rates, dropouts, and batch sizes.**

A black and white screen with numbers

AI-generated content may be incorrect.

**Table 2:** R2 performance of various ReLu activation functions, learning rates, dropouts, and batch sizes.

A black background with white numbers

AI-generated content may be incorrect.

The model below was built upon the highest performing r2 of the ReLu activation function combination. However, the R2 was different than above, likely due to seed initialization, which will be mentioned in the initial seed optimization section

# Design and Implementation Considerations

Training and experimentation were performed in Cursor on an ipynb notebook. Libraries used include TensorFlow 2.x, Scikit-learn, Pandas, NumPy, json, random, and matplotlib for analysis and visualization. Keras callbacks were instrumental in tuning learning schedules and tracking overfitting. Random Seed Optimization was performed to discover which seeds led to higher r2, and thus, lower MAE, RMSE, and MAPE.

# Results

An extended seed sensitivity analysis was conducted with 30 separate random seed initializations, and applied across different learning rates, dropout, and dense layer settings. This was done to better capture the interaction effects of hyperparameters with initialization randomness. The findings further reinforced the discovery that seed variance can dominate model behavior in smaller datasets, with R² varying as much as 0.12 between worst and best runs using the same architecture.

To visualize this, boxplots of R² scores across 30 seeds for each architecture were generated. The SARIMA-LSTM hybrid showed improved stability compared to vanilla LSTM, with a tighter range and consistently higher minimum performance. This supports the hypothesis that injecting SARIMA-informed seasonality acts as a regularizer for temporal noise.

Additionally, it was discovered that the use of summer binary indicators and sine/cosine encodings provided cyclical structure. This was directly useful when summer months (June through August) were weighed more heavily in training. Specifically, training with summer months having 1.5x or 2x sample weights yielded higher R² values during validation. This pattern

The results will compare various LSTM configurations using R-squared and RMSE on a holdout set. Additional plots and summary tables are forthcoming as experimentation concludes. The first model was constructed with optimization of evaluation ReLu and Sigmoid activation functions.

**Figure 3:** LSTM Model, random seed, with LSTM sequential modeling.

A graph with blue and orange dots

AI-generated content may be incorrect.

The next iterative process in the modeling was to apply a SARIMA-penalized model. This model performed slightly better, but also had variations based on seed initialization.

**Figure 4:** LSTM Model, random seed, with LSTM SARIMA Modeling.

A graph with blue and orange lines

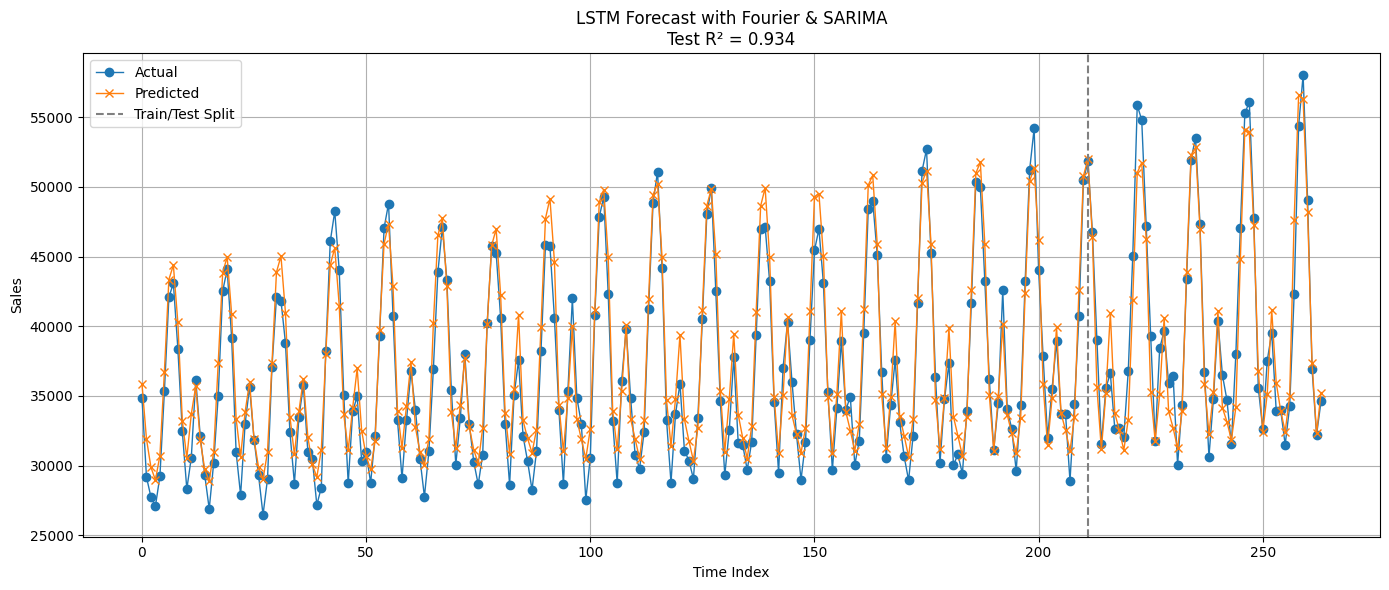
AI-generated content may be incorrect.

**Figure 5:** SARIMA LSTM Model, with optimized seed (50 iterations of picking highest r2 value seed and locking it).

A graph with blue and orange dots

AI-generated content may be incorrect.

**Figure 6:** This is the best current model, with a Fourier sin and cosine variables features and engineering a seed and hyperparameter tuning combination.



To address the high variance in LSTM model performance due to random seed initialization, a seed sensitivity analysis was conducted using 50 different seeds. Metrics evaluated included R², MAE, RMSE, and MAPE. The results demonstrated a nearly perfect negative correlation between R² and RMSE (-0.99), and strong correlations with MAE and MAPE (-0.97 and -0.94 respectively). This indicates that optimizing for R² also tends to improve other core forecasting metrics. The analysis underscores the critical impact of seed initialization in deep learning time series models, and the potential to use seed tuning as a lightweight optimization technique in practice. A correlation heatmap summarizing these findings is included below.

**Figure 7:** Correlation heatmap of R2, MAE, RMSE, and RMSE. This was derived from a 50 random seed initialization on the SARIMA-LSTM Model.

A red and blue squares with numbers

AI-generated content may be incorrect.

After this discovery was realized, the seeds were initialized for both the SARIMA-LSTM Model, and the LSTM Model. It can be noted that the SARIMA LSTM has a higher average r2, as well as tighter spread amongst the randomized seeds.

**Figure 8:** This graph displays the R2 of an LSTM model, with 30 random initialized seeds, and scored an average of .765.

A graph with a red line

AI-generated content may be incorrect.

**Figure 9:** This graph displays the R2 of an LSTM model, with 30 random initialized seeds, and scored an average of .856.

A graph of blue and red lines

AI-generated content may be incorrect.

# Analysis and Interpretations

This LSTM model has been architected to fit the EIA dataset extremely well and should be able to be extrapolated to future energy levels in the residential sector, especially given just over 200 data points for training. The starting process of the model architecture was to engineer various features, through EDA. The main variable of interest was the summer encoded variable. Then, it was evaluated how these features fit to the baseline LSTM model, which was the model chosen to fit to this problem due to the sequential nature of the problem, and how the relevance is year over year growth is prevalent. The initial parameters were inserted into various combinations to determine the optimal parameters to continue to the next step, which was optimizing initial seeds, through over 200 iterations. Next, parameter tuning was run again to have an updated baselevel and seeing which hyperparameters fit the seed the best. While there is currently no package that can automate this process, this was extremely helpful to re-evaluate hyper-parameters after an initial seed was found.

When models are compared, this paper suggests a new framework that there should be a weighted average of at least 30 seeds (adequate sample size) to get a realistic comparison and adjust for randomness. As (Picard, 2021) demonstrates, the choice of random seed can significantly influence deep learning results “I strongly suggest aspiring authors to perform a randomness study by varying seeds - and if possible dataset splits - and reporting average, standard deviation, minimum and maximum scores.”

# Conclusions

By integrating SARIMA-based seasonal adjustment with LSTM and carefully tuning seed initialization, this study demonstrates a scalable framework for energy demand forecasting. The stability of the SARIMA-enhanced model under multiple random seeds confirms its practical robustness to changing initializations.

The feature set, particularly the cyclical and customer-derived variables with Fourier sin and cosine transformation, enabled the model to generalize across regional and sector demand fluctuations. The seed analysis revealed that, despite deep learning’s stochastic nature, initialization can be controlled to ensure repeatability and reproducibility.

This lays the foundation for future hybrid pipelines where temporal priors (SARIMA, Fourier terms) are fused with modern architectures such as attention-based LSTMs or TCNs to further enhance interpretability and accuracy in energy time series modeling.

The LSTM model, adjusted with SARIMA-based summer weightings, demonstrated strong predictive performance. The best configuration achieved an R² of 0.934, with consistent behavior across other evaluation metrics. The seed sensitivity analysis confirms that initialization significantly impacts performance, but careful hyperparameter and weight handling can mitigate this and even improve performance. The pipeline provides a robust template for seasonal energy forecasting at scale and could be utilized in the future to accurately predict energy sales.

# Future Direction

Another addition to this paper would be to apply 30 different random seeds to each learning rate, dense layer, both dropouts.

Future work should integrate exogenous weather variables and grid-level macroeconomic indicators. Expanding the SARIMA adjustment into non-linear embeddings or attention mechanisms could improve robustness. Alternative deep architecture such as Temporal Convolutional Networks or Transformers are next steps that could be explored more in depth.

Additionally, further work could involve going into SARIMA fine tuning, and looking into time steps for the features added, for future pattern detection. Also, it is possible that other time series related feature engineering would be more precise for the dataset. As well as the potential for cross-validation to other models to validate the model selection for this data set. It could also be useful to break the modeling down into various states for prediction and evaluate the error rates for each state, as well as sector. The data set was focused just on the residential sector, but there is room to evaluate industrial and commercial as well, as well as three in conjunction.

Further work could be done to ensure the model is robust, such as evaluating autocorrelation, AIC and BIC evaluation, and to build these into the model. Likewise, covariation and deeper studies into batch norm could prove useful.

# Bibliography

[1] Choi, Eunjeong, Soohwan Cho, and Dong Keun Kim. 2020. “Power Demand Forecasting using Long Short-Term Memory (LSTM) Deep-Learning Model for Monitoring Energy Sustainability.” Sustainability 12, no. 3: 1109. https://doi.org/10.3390/su12031109.

[2] Emshagin, Saad, Wayes Koroni Halim, and Rasha Kashef. 2022. “Short-Term Prediction of Household Electricity Consumption Using Customized LSTM and GRU Models.” arXiv, December 16, 2022. https://arxiv.org/abs/2212.08757.

[3] Hochreiter, Sepp, and Jurgen Schmidhuber. 1997. “Long Short-Term Memory.” Neural Computation 9(8) (1735-1780): 1-32. https://www.bioinf.jku.at/publications/older/2604.pdf.

[4] Hrnjica, Bahrudin, and Ali Danandeh Mehr. 2019. “Energy Demand Forecasting Using Deep Learning.” In Smart Cities, Performability, Cognition, & Security pp 71-104. EAI/Springer Innovations in Communication and Computing. Cham: Springer International Publishing. doi:10.1007/978-3-030-14718-1\_4.

[5] Kaya, Mahmut, Anıl Utku, and Yavuz Canbay. 2024. “A Hybrid CNN-LSTM Model for Predicting Energy Consumption and Production Across Multiple Energy Sources.” Journal of Soft Computing and Artificial Intelligence 5, no. 2: 63–73. <https://doi.org/10.55195/jscai.1577431>.

**[6] Kaggle*.*** House Prices: Advanced Regression Techniques **competition dataset.** Accessed July 23, 2025. <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques>.

[7] Kaur, Devinder, Shama Naz Islam, Md. Apel Mahmud, Md. Enamul Haque, and Zhao-Yang Dong. 2020. “Energy Forecasting in Smart Grid Systems: A Review of the State-of-the-Art Techniques.” arXiv, November 25, 2020. <https://arxiv.org/abs/2011.12598>.

[8] Picard, David. “Torch.manual\_seed (3407) Is All You Need: On the Influence of Random Seeds in Deep Learning Architectures for Computer Vision.” *arXiv e‑print*, submitted September 16, 2021; revised May 11, 2023 (v2). arXiv:2109.08203. https://doi.org/10.48550/arXiv.2109.08203

[9] Polson, Michael, and Vadim Sokolov. 2018. “Deep Learning for Energy Markets.” arXiv, August 16, 2018. <https://arxiv.org/abs/1808.05527>.

[10] Ren, Xiaohang, Weixi Xu, and Kun Duan. 2022. “Fourier Transform Based LSTM Stock Prediction Model under Oil Shocks.” *Quantitative Finance and Economics* 6 (2): 342–58. <https://doi.org/10.3934/QFE.2022015>

[11] Roy, Koushik, Abtahi Ishmam, and Kazi Abu Taher. 2021. “Demand Forecasting in Smart Grid Using Long Short-Term Memory.” arXiv, July 28, 2021. <https://arxiv.org/abs/2107.13653>.

[12] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” *Journal of Machine Learning Research* 15 (1): 1929–1958. <https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>.