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

Ethan Norton [1], †

[1]Northwestern University School of Professional Studies

Master of Science in Data Science Program

633 Clark St, Evanston, IL 60208

†Address to which correspondence should be addressed:

EthanRNorton@protonmail.com

Revision Date: August 11th, 2025

Abstract

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. Reliable resident electricity forecasting is crucial as rapid adoption enhances demand on energy infrastructure. However, precise forecasting models can inform utility planning, optimize distribution, and improve efficient cost structures for providers and consumers. With that being said, 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 R^2, 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

Table of Contents

Introduction and Problem Statement

The stability and consistency of residential energy supply is now tightly correlated with the growth of AI and large-scale computing, such as LLMs; which can introduce unpredictability and seasonal or load based spikes to the grid. Therefore, robust and precise prediction is imperative for utility companies, policymakers, and energy markets to allocate resources effectively. 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 with one of the most important (yet overlooked) aspects of enhancing LSTM performance is 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.

Additionally, traditional statistic models such as ARIMA models, moving averages, and linear regression provide interpretable baselines by assume linearity and stationarity. This assumption violates the non-linear and cyclical patterns of modern energy consumption. To address these downsides, this study integrates the SARIMA-informed seasonality blended into the LSTM framework, emphasizing and building the strengths of both statistical and deep learning approaches. An additional contribution is systematic exploration of initial seed optimization, where 50 random initializations displayed that seed choice can drastically alter results. By averaging and strategically selecting optimized seeds, the model achieves improved accuracy, stability, and reproducibility,

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, Hrnjica and Mehr (2019) investigated the process of training the LSTM deep learning model, which is a subset of the RNN model. 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 as showcased by 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.

Parameter tuning was optimized and evaluated to architect a more precise model, and to build a baseline of what was explored in order studies, and then to further implement improvements. 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.

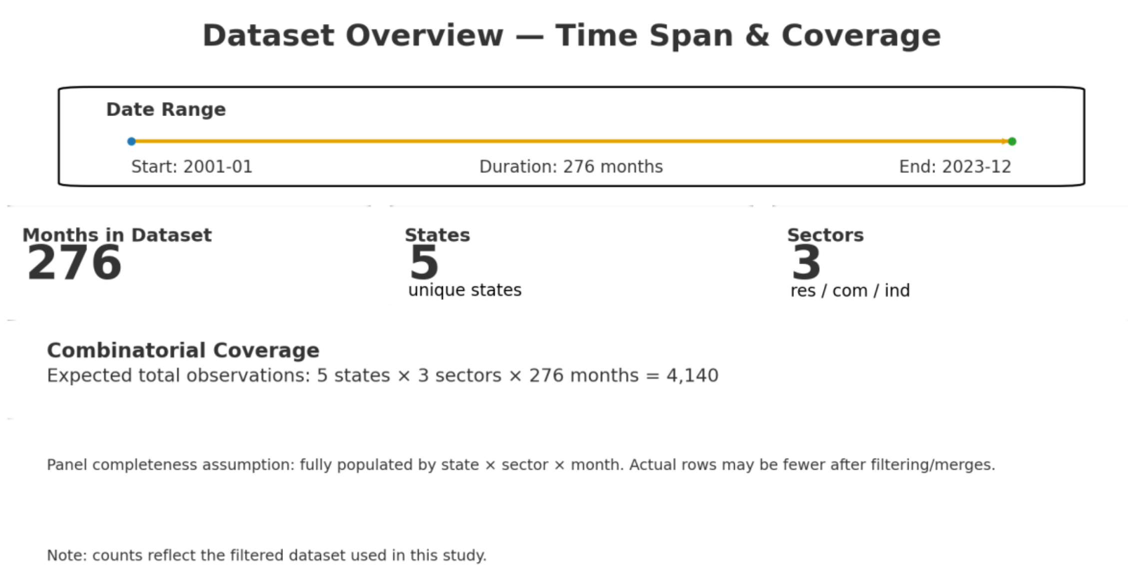
One paper, Roy, Ishmam, and Tahe, (2021) evaluated the application of many deep learning methodologies – both deterministic and probabilistic, and emphasized the potential for hybrid models "Statistical methods face challenges from their inability to process big data, while AI-based methods have other 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.



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.
- **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.

SARIMA-informed weighting for LSTM

A SARIMA forecaster was not trained alongside or in conjunction with the LSTM. Instead, we borrow SARIMA's core principles, explicit treatment of seasonal structure and Box Jenkins diagnostics, to create a seasonal prior that regularizes the neural model. With monthly data (seasonal period $s = 12$). The classical SARIMA model (shown below for context) is not used directly for prediction in our pipeline.

Rather than using a full SARIMA fit for forecasts, we convert year-over-year consumption growth into a smooth seasonal signal and map it into (0,1) with a logistic ("sigmoid") transform to obtain a bounded prior. Sample weights were composed, as well as s summer encoded variable, that modestly boost high-signal months and gently suppress low-signal months, followed by clipping and normalization so the average weight is 1. These weights are passed as sample_weight during LSTM training on sliding windows. This SARIMA-informed seasonal regularization drastically improves average accuracy (higher R^2, lower RMSE/MAE) and reduces seed-to-seed variance by encouraging seasonally aware representations, as demonstrated later.

Additionally, this study includes a separate summer weighting factor based on the empirical average consumption differential between summer and non-summer months. Together, the SARIMA-informed prior and the seasonal factor serve as light-touch, domain-aware regularizer that shapes what the LSTM learns, without hard-coding explicit lags or differencing. Figure 2: SARIMA implementation, and model layout.

Monthly energy use is driven by recurring climate and behavior cycles; summer cooling loads, winter heating, holiday closures, daylight shifts, and rate changes.

The SARIMA informed prior isolates this recurring structure by smoothing YoY growth for each calendar month, so it tracks how, for example, July behaves relative to last July while remaining robust to long-run level shifts (customer count, pricing, economic drift). Combining that prior with a calibrated summer factor yields sample weights that gently up-weight months where the system is most volatile and operationally critical. The LSTM then trains with these weights, so its loss emphasizes learning the recurring peaks and troughs while still capturing nonlinear interactions among covariates (e.g., 5 customers, price, temperature proxies) and any residual dynamics left unmodeled by SARIMA principles.

This tends to raise R-squared for two practical reasons. First, peak months contribute a large share of the total variance; reducing errors at peaks disproportionately improves R-squared even if average months change little. Second, the weighting acts as seasonal regularization, which stabilizes training across random seeds and narrows the spread of outcomes. This report evaluates an unweighted chronological split to ensure the gains reflect genuine generalization rather than a metric artifact.

Research Design and Modeling Method(s)

This project uses a multivariate LSTM modeling approach implemented in TensorFlow/Keras. The model architecture includes:

- One or more LSTM layers, which ranged from 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.

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.

Figure 2: ReLu and Sigmoid Activation Functions, and various outcomes for hyperparameter tuning.

Activation	Top R^2
ReLU	0.864
Sigmoid	0.777

Rank	R^2	Activation	lr	LSTM (h1,h2)	Dense	Dropout
#1	0.864	ReLU	0.0005	(64, 64)	16	0.2
#2	0.751	ReLU	0.001	(128, 64)	32	0.2
#3	0.693	ReLU	0.001	(64, 64)	32	0.3

Notes: lr = learning rate. LSTM (h1,h2) lists hidden sizes for the two stacked LSTM layers. Dropout refers to the main dropout applied in the recurrent stack.

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 R^2, and thus, lower MAE, RMSE, and MAPE.

Interpretations

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 R^2 varying as much as 0.12 between worst and best runs using the same architecture.

To visualize this, boxplots of R^2 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 R^2 values during validation. This pattern is apparent in the improved performance in the SARIMA LSTM model.

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

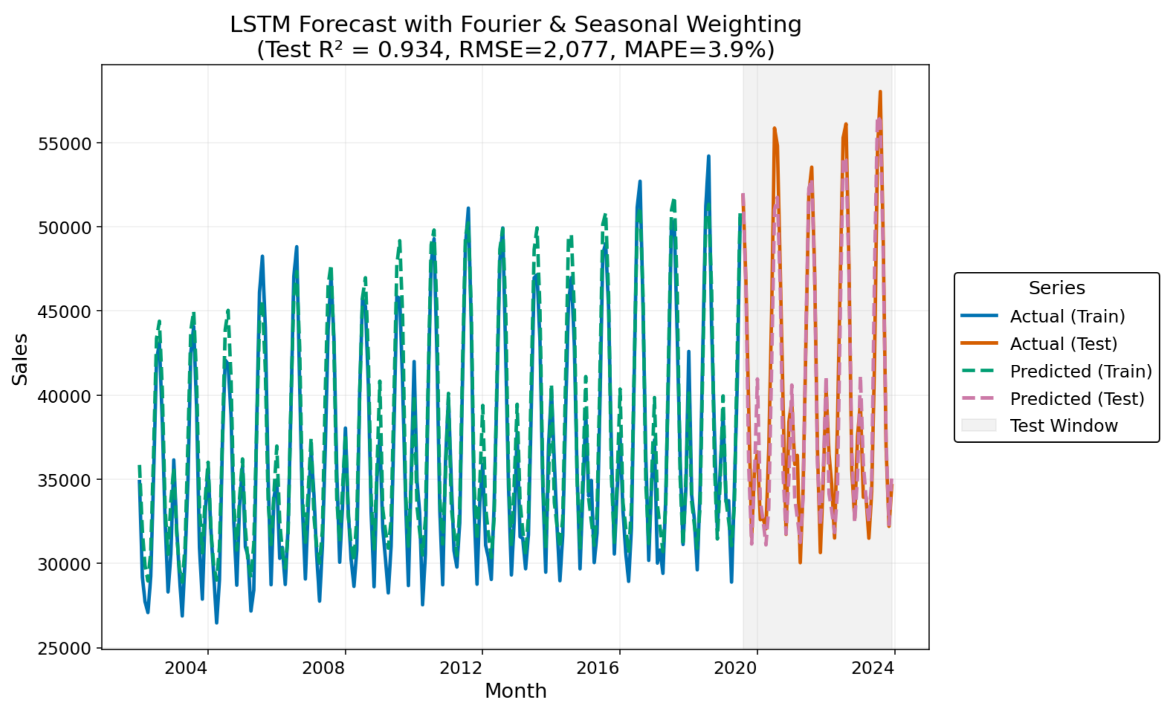


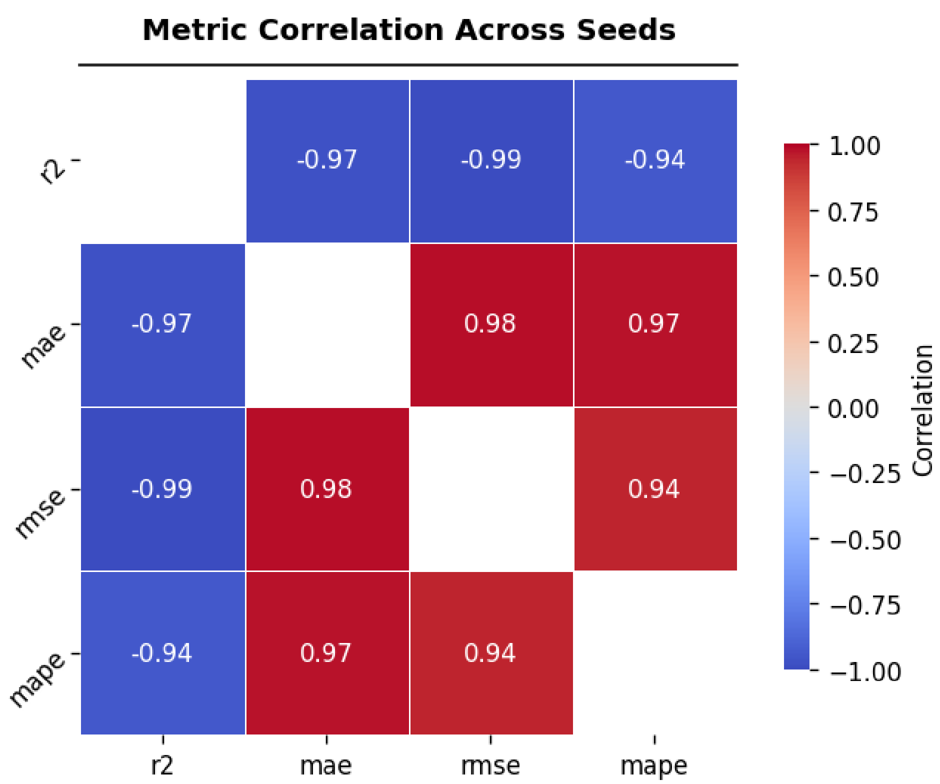
Fig. 3A. Solid lines = actuals; dashed = predictions. Shaded region is out-of-sample test.

The chart compares monthly actuals (solid) and model outputs (dashed) for train and for the shaded out-of-sample test window. The LSTM with Fourier month encodings and SARIMA-informed seasonal weighting closely tracks the timing and amplitude of the seasonal cycle and the gradual level growth in the series. In the test period, the forecast (pink dashed) overlays the actuals (orange) with **R^2 = 0.934**, **RMSE ≈ 2,077**, and **MAPE**

≈ 3.9%, indicating high explanatory power and low relative error. Residual discrepancies are mainly at the sharpest seasonal peaks and shoulder transitions; baseline months show minimal bias. The tight alignment and lack of visible drift across the test horizon suggest stable generalization and support the effectiveness of injecting a bounded seasonal prior into the LSTM.

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 R^2, MAE, RMSE, and MAPE. The results demonstrated a nearly perfect negative correlation between R^2 and RMSE (-0.99), and strong correlations with MAE and MAPE (-0.97 and -0.94 respectively). This indicates that optimizing for R^2 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 4: Correlation heatmap of R^2, MAE, RMSE, and RMSE. This was derived from a 50 random seed initialization on the SARIMA-LSTM Model.



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 R^2, as well as tighter spread amongst the randomized seeds.

Figure 5: This graph displays the R^2 of an LSTM model, with 30 random initialized seeds, and scored an average of .765.

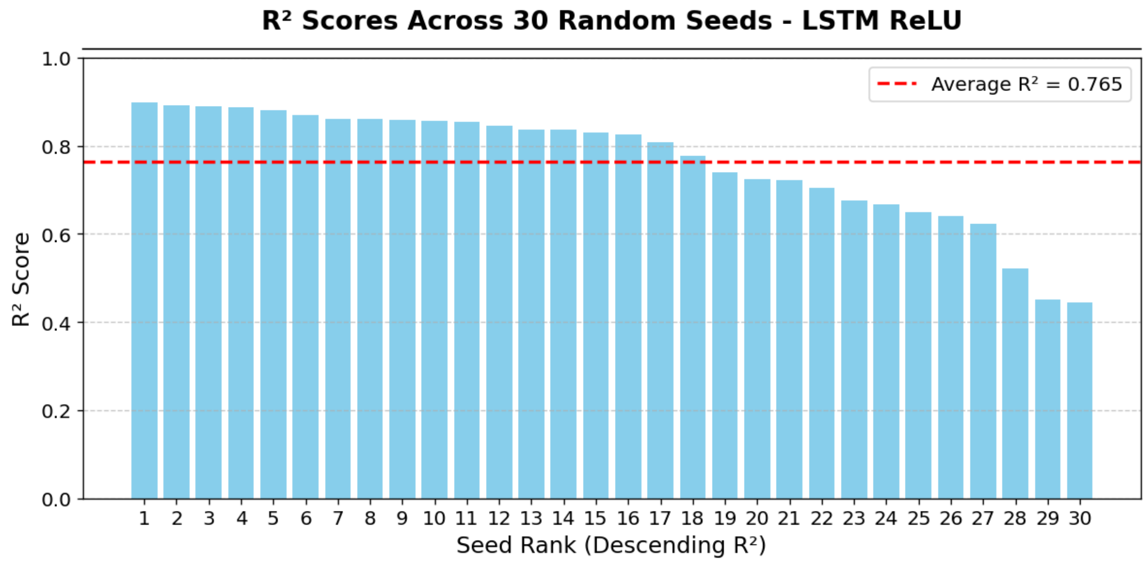
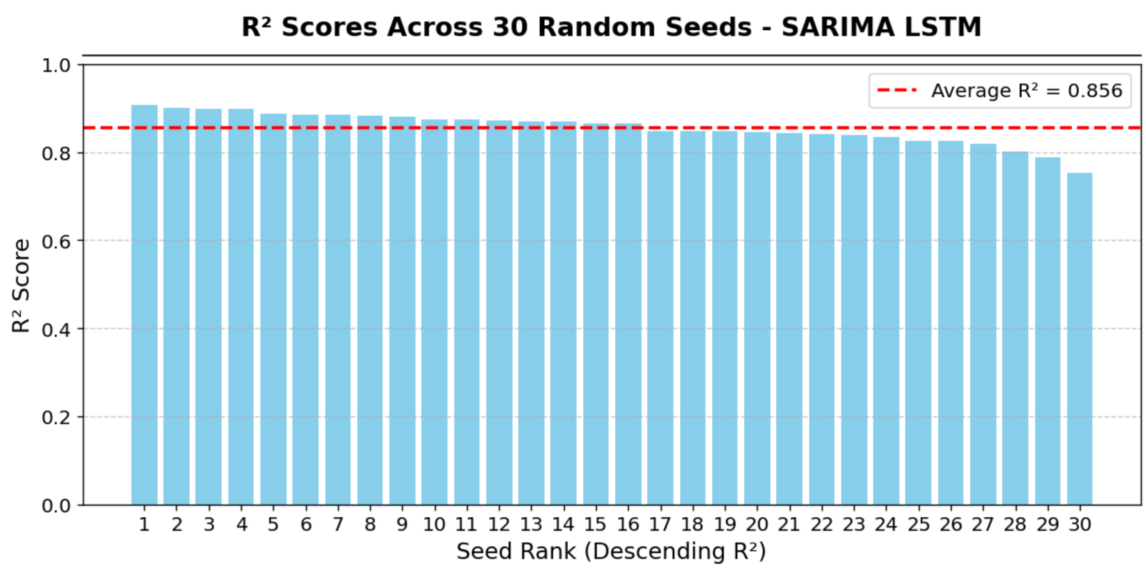


Figure 6: This graph displays the R^2 of an LSTM model, with 30 random initialized seeds, and scored an average of .856.



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."

Results

The evaluation compared the baseline LSTM model and the SARIMA-informed LSTM variant across 30 random seed initializations, with additional experiments exploring hyperparameter combinations including dropout rates, learning rates, and dense layer sizes. Performance was assessed using R^2, RMSE, MAE, and MAPE on the holdout set.

Across all configurations, the SARIMA-informed LSTM consistently outperformed the baseline LSTM in both predictive accuracy and stability. The average R^2 for the baseline LSTM was **0.765**, compared to **0.856** for the SARIMA-informed variant. Moreover, the SARIMA-informed model exhibited a narrower interquartile range in R^2 values across seeds, indicating reduced sensitivity to initialization randomness. This regularizing effect was particularly evident in summer peak months, where SARIMA-derived seasonal weightings provided measurable gains in capturing cyclical demand patterns.

The seed sensitivity analysis revealed that model initialization could cause R^2 values to vary by up to **0.12** within the same architecture and hyperparameter set. A strong negative correlation ($r = -0.99$) was observed between R^2 and RMSE, with similarly strong correlations for MAE (r = -0.97) and MAPE (r = -0.94). This suggests that improving R^2 through seed and hyperparameter tuning reliably improves other core error metrics.

Figure 7: Summary of the Results of the two various models, as displayed in Figures 5 and 6.

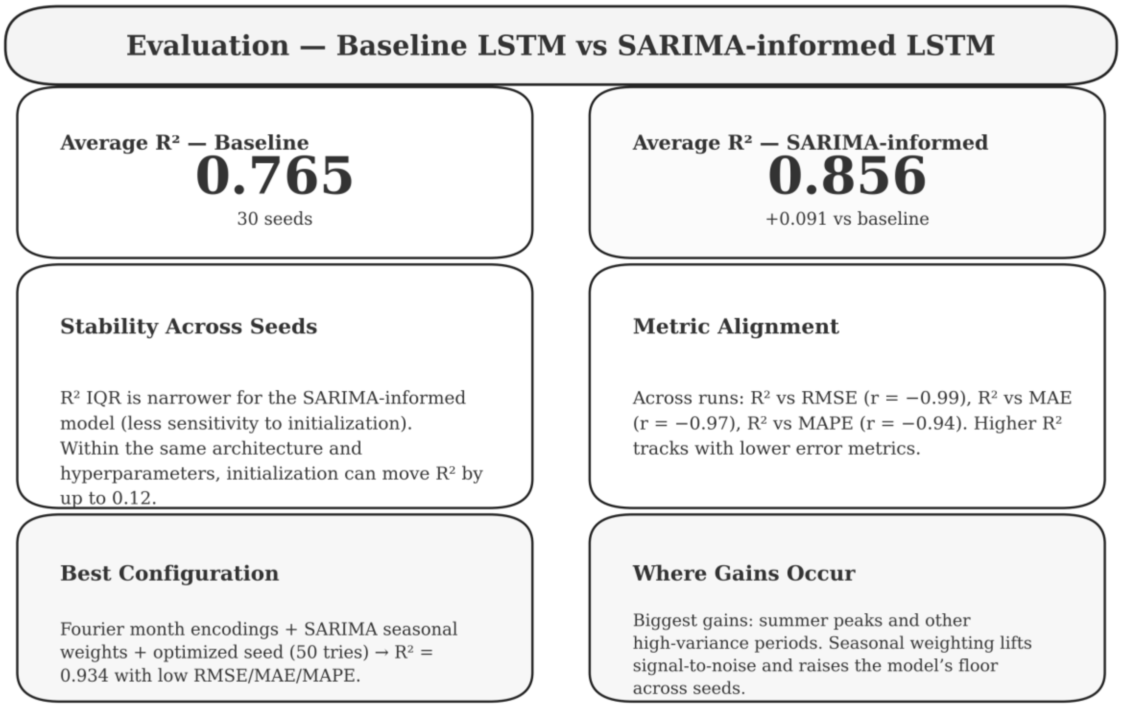


Figure 8 and Figure 9 illustrate the R^2 distributions for the baseline and SARIMA-informed models, respectively. While both models achieved high-accuracy runs, the SARIMA-informed model demonstrated a higher floor performance and more consistent behavior. The best configuration, incorporating Fourier-based cyclical encodings, SARIMA seasonal weightings, and an optimized seed from 50 iterations, achieved an R^2 of **0.934** with corresponding low error values across all metrics.

These results support the conclusion that seasonal signal injection, when applied through domain-informed feature engineering, improves both the accuracy and reproducibility of deep learning models for residential energy demand forecasting.

Conclusions

- Accuracy improves: Average $\mathbf{R^2}$ **0.765 → 0.856** (**+0.091**). Errors (RMSE/MAE/MAPE) fall accordingly.

- Stability improves: The SARIMA-informed model shows a **narrower R^2 interquartile range** and higher floor across 30 seeds.

- Metric alignment: R^2 is **strongly negatively correlated** with RMSE/MAE/MAPE; when R^2 rises, all three-error metrics drop.

- Where gains occur: **Largest lifts in summer/peak months**, where seasonal weighting raises signal-to-noise.

- Seed sensitivity: Within a fixed architecture, **initialization can shift R^2 by ~0.12**; SARIMA weighting reduces this volatility.

- Best run: **R^2 = 0.934** using Fourier months + SARIMA weights + optimized seed (50 iterations), with low RMSE/MAE/MAPE.

- Takeaway**:** A **light seasonal prior** regularizes the LSTM without hard-coding lags/differencing, improving accuracy and reproducibility.

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 R^2 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

The Introduction frames a practical need: utilities must anticipate seasonal peaks and emerging demand patterns to protect reliability, ratepayers, and decarbonization goals, while maintaining an infrastructure for LLMs and data centers. This study showed that seasonality-aware LSTM, regularized by a SARIMA-informed prior, improves accuracy and training stability on monthly residential load. The next step is to turn this into an operational, generalizable forecasting stack that remains trustworthy under changing weather regimes, policy shocks, and regional heterogeneity. Large Language Models (LLMs) can accelerate that transition by ingesting unstructured context (operator notes, bulletins, policy updates), automating experiment orchestration, and translating technical results for non-technical decision makers.

- **Reproducibility & Seed Robustness.**
  Apply **30 random seeds per** learning rate, dense-size, and dropout combination; summarize distributions (median/IQR) rather than single runs. Integrate residual whiteness checks and information-criteria reporting (AIC/BIC) directly into the training pipeline.
- **Exogenous Drivers.**
  Incorporate **weather** (HDD/CDD, heat-wave flags, ENSO/heat index), **grid-level macro indicators** (economic activity, rates, policy), and tariff/price signals. Evaluate non-linear embeddings and light attention to fuse these drivers with the seasonal prior.
- **SARIMA Refinement.**
  Systematically **fine-tune seasonal orders** and differencing; test sensitivity to **time steps** used in derived features. Compare the current smoothed prior to **learned seasonal embeddings** and to Fourier-only baselines.
- **Architectures & Model Selection.**
  Benchmark **Temporal Convolutional Networks** and **Transformers** against the SARIMA-informed LSTM. Use rolling-origin cross-validation and compare to classical baselines (ETS/ARIMA/Prophet) and modern tree or hybrid models (e.g., XGBoost w/ lags) to validate model choice.
- **Regional & Sectoral Generalization.**
  Break out models by **state** and **sector** (residential, commercial, industrial), report error by region/sector, and assess transfer learning between jurisdictions. This extends beyond the current residential focus.
- **Feature Engineering Depth.**
  Explore additional time-series features (holiday effects, rate-change indicators, outage/DER adoption flags). Test alternative smoothing and clipping strategies for the seasonal weights to balance bias/variance.
- **LLM-Enabled Workflow Enhancements.**
  Use LLMs to:
  • **Normalize and audit** metadata, logs, and operator notes;
  • **Auto-generate scenarios** ("hot/dry summer + price spike") that condition exogenous features for stress tests;
  • **Orchestrate ablation grids** (code generation for seed/LR/dropout sweeps) and

produce **experiment summaries**;
• **Summarize and explain** model behavior and uncertainty for planners/executives;
• **Retrieve** recent policy/weather advisories (RAG) and convert them into structured features.

- **Deployment Readiness.**
  Add calibrated **uncertainty quantification**, concept-drift monitoring, and guardrails for regime shifts. Package the seasonal prior + LSTM as a reproducible service with automated reports and audit trails.

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

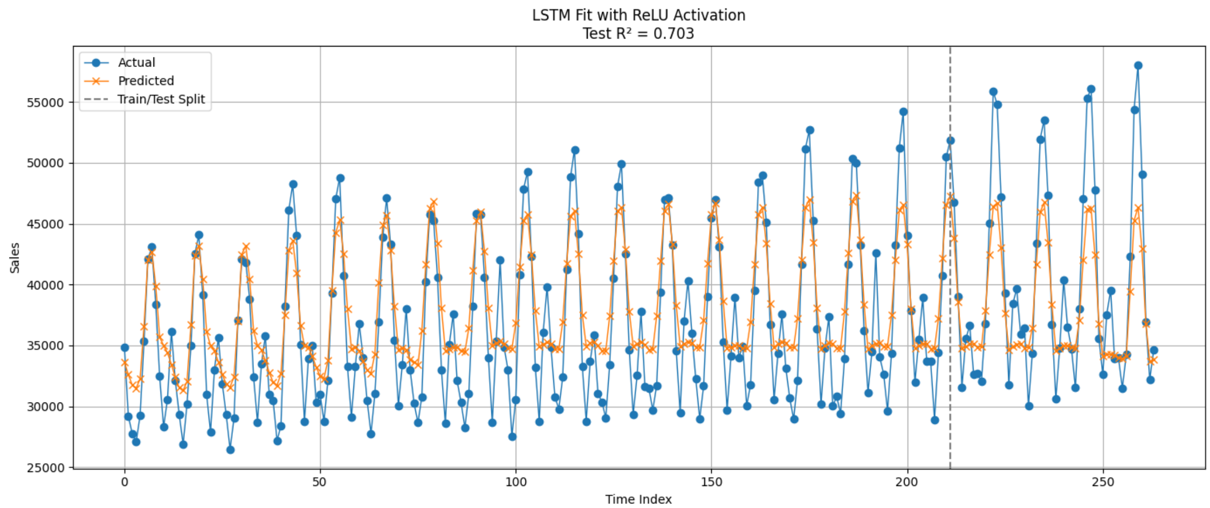
[11] Roy, Koushik, Abtahi Ishmam, and Kazi Abu Taher. 2021. "Demand Forecasting in Smart Grid Using Long Short-Term Memory." arXiv, July 28, 2021. https://arxiv.org/abs/2107.13653.

[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.

Appendix

Figure 8: LSTM Model, random seed, with LSTM sequential modeling.



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 9: LSTM Model, random seed, with LSTM SARIMA Modeling.

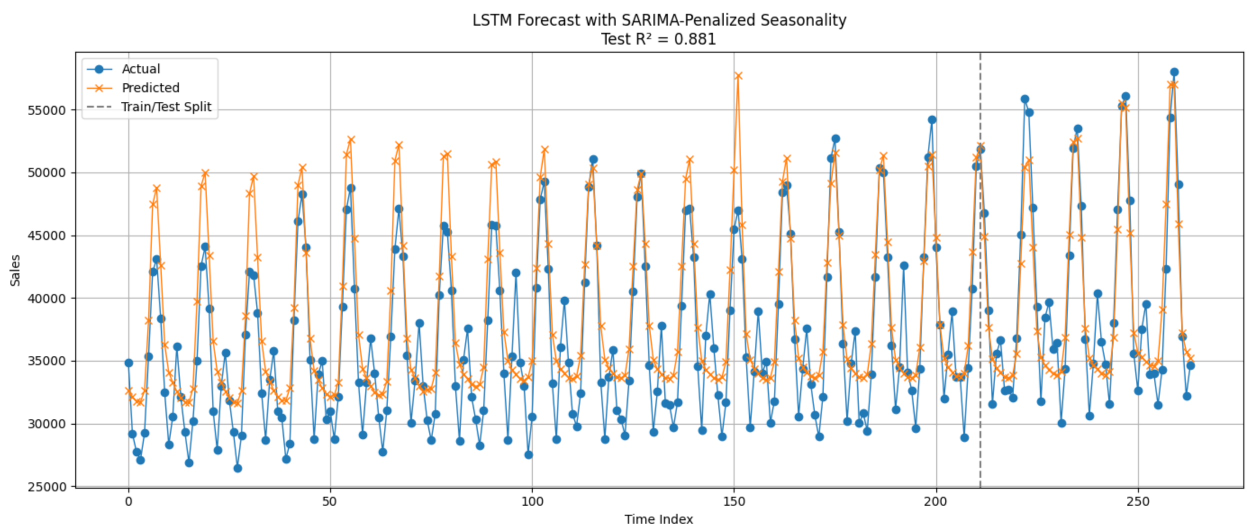


Figure 10: SARIMA LSTM Model, with optimized seed (50 iterations of picking highest R^2 value seed and locking it).