Forecasting Electricity Usage: Comparative Analysis of Prophet, LSTM, and Hybrid Models on Public Datasets

Tyler James, R01611773
Artificial Intelligence Management 460
Farmingdale State College
2350 NY-110, Farmingdale, New York, United States, 11735
Email: Jamet@farmingdale.edu

October 22, 2025

Abstract

Short-term electricity load forecasting (STELF) is important for energy allocation, grid stability, and operating cost. This paper compares three models: Facebook Prophet (Prophet), a Long Short-Term Memory (LSTM) network, and a Hybrid that trains an LSTM on Prophet residuals. For this paper and comparative, three public datasets with different characteristics were chosen: volatile household power, seasonal aggregate load diagrams, and smoother electricity transformer oil temperature. The models were evaluated by using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The Hybrid model had the lowest error on the two seasonal datasets because Prophet captures trend and seasonality while the LSTM corrects non-linear deviations. The LSTM model is best on the smoother Electricity Transformer Temperature (ETT) dataset because Prophet's structure is less suitable and the Hybrid model did not perform as well as LSTM. [1-7]

1 Introduction

Reliable load forecasting is key for running a stable power grid, because it helps to balance energy supply with energy demand, while also helping to manage daily operations. Predicting usage hours in advance (short-term) is key for scheduling at power plants. This task is becoming harder and harder as more renewable energy is used and demand patterns become less predictable. Older statistic models can handle simple seasonal values but fail when the data has complex changes. Deep learning models, especially LSTMs (Deep Learning Models), are better for this because they can learn long-term patterns and non-linear trends found in energy data. Facebook Prophet (a Statistical Model) is another tool used as a strong starting point (baseline). It breaks the data down into main trend, seasonal values (ex. monthly, weekly or daily), and holiday effects. Studies show that Prophet can perform well, especially in national or building-level forecasts where demand patterns are important. This project compares Prophet against LSTM to see which model performs better on short-term electricity forecasts. It also tests a Hybrid model that combines both: it runs Prophet first, then trains an LSTM on Prophet's mistakes (the residuals), and adds the corrections back to Prophet's main forecast [1-6].

2 Related Work

Recent studies show preprocessing, model structure, external data features (ex. Weather, holidays, weekends, etc.), and evaluation for deep-learning based short-term load forecasting, need backtesting and data leakage safe validation. Prophet versus LSTM comparisons show that the outcomes are dependent on the datasets and problems. A national demand study selected Prophet as the most accurate and most interpretable for policy scenarios, while a digital-twin study found Prophet competitive for one day forecasts in energy where calendar signals are dominate. Hybrid models combine Prophet with LSTM. Bashir et al. fit Prophet, trained an LSTM on the residuals, and reduced errors on Elia grid data. Arslan separated seasonality with Prophet and modeled the remaining parts with stacked BiLSTM. Albahli used dynamic weighting between LSTM and Prophet to improve hourly demand accuracy. Yang et al. used temporal components (overall trend, weekly/yearly seasonality, and holiday effects) with weather features using a ConvLSTM-3D network. The findings in these papers suggest hybrids help when seasonality is strong but local deviations remain, where end to end LSTMs do better with smoother autoregressive signals. [1-6]

3 Datasets and Experimental Setup

3.1 Datasets

Three datasets commonly used in energy forecasting research were selected for their differing characteristics:

1. **Household Power Consumption (UCI):** Minute-level household measurements resampled to hourly, with the target being Global Active Power while the series is volatile with spikes. Target: `Global_active_power`, resampled to hourly.

2. **Electricity Load Diagrams (UCI):** Regional aggregate load with strong daily and weekly values that have a clear seasonal structure. Target: Summed `total_consumption`, resampled to hourly.

3. **Electricity Transformer Temperature (GitHub):** Industrial sensor temperature series with smoother autoregression; used in many recent ConvLSTM/feature-fusion studies. [1]. Target: `OT`.

All of the datasets went through preprocessing which included date parsing, handling missing values (e.g., imputation for Household data, removal of zero-load hours for Load Diagrams), and resampling to hourly frequency. Each dataset was then split into: 80% training and 20% testing. Dataset details after preprocessing are in Table 1. For compatibility with Prophet, all datasets were formatted into dataframes with `ds` (datetime) and `y` (target value) columns.

Table 1: Dataset Summary After Preprocessing

Dataset	Source	Target Variable	Orig. Freq.	Final Freq.	Time Period	Train Pts	Test Pts	Total Pts
1: Household Power	UCI	`Global_active_power`	Minute	Hourly	Dec '06–Nov '10	27671	6918	34589
2: Load Diagrams	UCI	`total_consumption`	15-Min	Hourly	Jan '11–Dec '14	28052	7013	35065
3: ETT (Oil Temp)	GitHub	`OT`	Hourly	Hourly	Jul '16–Jun '18	13936	3484	17420

3.2 Evaluation Metrics

Model accuracy was evaluated using regression metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), calculated on test set predictions against true values. MAE was used as the primary metric for comparing model performance because of its direct interpretability.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_{\text{true}} - y_{\text{pred}}| \tag{1}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{\text{true}} - y_{\text{pred}})^2} \tag{2}$$

Calculations were performed using `scikit-learn`'s `mean_absolute_error` and `mean_squared_error` functions.

4 Methodology

The Prophet, LSTM, and Hybrid models were implemented using libraries such as `pandas` for data manipulation, `prophet` for the Prophet model, `tensorflow.keras` for LSTM implementation, and `scikit-learn` for data scaling and metric calculation.

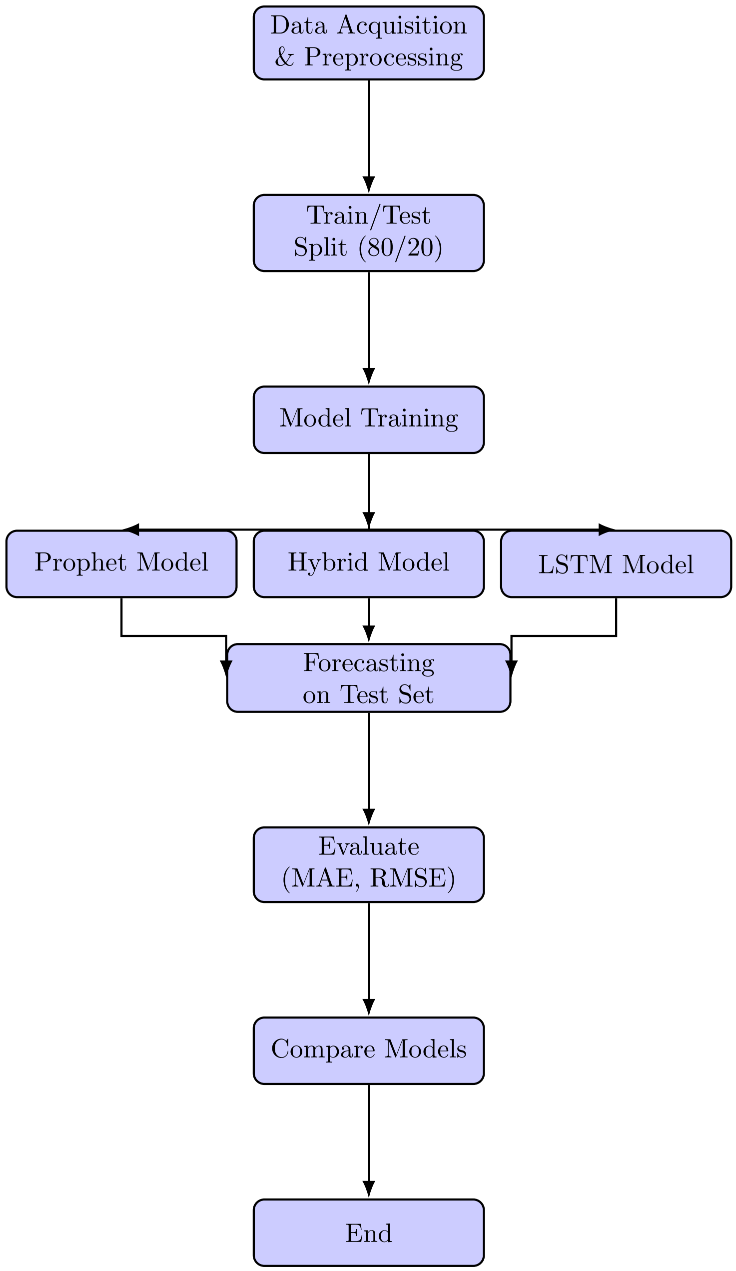

Figure 1: Overall Methodology Flowchart

4.1 Prophet Model

The Facebook Prophet library was useded. Prophet models time series $y(t)$ though an additive: $y(t) = g(t) + s(t) + h(t) + \epsilon(t)$, where $g(t)$ is the trend, $s(t)$ for seasonality, $h(t)$ for holiday effects, and $\epsilon(t)$ is error [2, 3]. [1, 4]. The model object was instantiated, fitted (`.fit()`) on the training portion (`train_df`) of each dataset, and then used to generate forecasts for the timestamps corresponding to the test set using `make_future_dataframe` and `.predict()`.

4.2 LSTM Model

The LSTM network predicts the next value from fixed-length histories (one week) using a single LSTM layer followed by an output layer, trained with mean-squared error and early stopping. LSTMs capture long-range dependencies which are common in electricity demand sequences. [2, 5].

1. **Scaling:** Input data (`y`) was scaled to the [0, 1] range.
2. **Sequencing:** A sequence length (`SEQUENCE_LENGTH`) of 24 hours was chosen, meaning the model used the previous days hourly values to predict the value for the next hour.
3. **Reshaping:** Data was shaped into `[samples, timesteps=24, features=1]` required by Keras LSTM layers.
4. **Training:** The model was put together using the `adams` optimizer and `mean_squared_error` as the loss function. Training had 10 epochs using a batch size of 32 on training sequences.
5. **Prediction & Inverse Scaling:** The trained model generated predictions on test sequences, which were inverse-scaled back to the original range for evaluation.

Hyperparameters are listed in Table 2.

Table 2: Key LSTM Hyperparameters

Parameter	Value
LSTM Units	50
Sequence Length	24 hours
Optimizer	Adam
Loss Function	Mean Squared Error (MSE)
Epochs	10
Batch Size	32

4.3 Hybrid (Prophet + Residual LSTM) Model

The hybrid model combined Prophet's model trend and seasonality with LSTM's complex pattern capturing ability, similar to [2, 4]. The main idea is to use the Prophet model for the main structure and then use LSTM to catch the residuals left by Prophet [2].

1. Generate a base forecast using the trained Prophet model.
2. Calculate the residuals by subtracting the Prophet forecast from the actual values.
3. Train a LSTM model (using the same process as the solo LSTM) on the residuals in the training data.
4. Use the trained residual LSTM to predict residuals for the test data.
5. Compute the final hybrid forecast by adding the predicted residuals to the Prophet baseline forecast: $yhat_{\text{Hybrid}} = yhat_{\text{Prophet (test)}} + residuals_{\text{LSTM (test)}}$.

5 Results

Table 3 presents the MAE results for Prophet, LSTM, and the Hybrid model on the test set for each of the three datasets. The best performing model for each dataset is highlighted, along with percentage improvements relative to Prophets baseline.

Table 3: Final Performance Summary (MAE, lower is better)

Dataset	Prophet MAE	LSTM MAE	Hybrid MAE	Best Model	Δ vs Prophet (LSTM)	Δ vs Prophet (Hybrid)
1: Household Power	29.95	22.77	**22.35**	**Hybrid**	23.96%	25.35%
2: Load Diagrams	128362.52	29959.26	**24913.74**	**Hybrid**	76.66%	80.59%
3: ETT (Oil Temp)	12.66	**0.47**	0.89	**LSTM**	96.29%	92.97%

Note: MAE values from experimental results. Improvements (Δ) calculated as $(1 - \textit{Model MAE}/\textit{Prophet MAE}) \times 100\%$.

5.1 Discussion by Dataset

5.1.1 Dataset 1: Individual Household Electric Power Consumption

The Hybrid model had the lowest MAE, it improved Prophet by nearly a quarter; plots show Prophet captured shape while the residual LSTM corrected the peak underestimations.

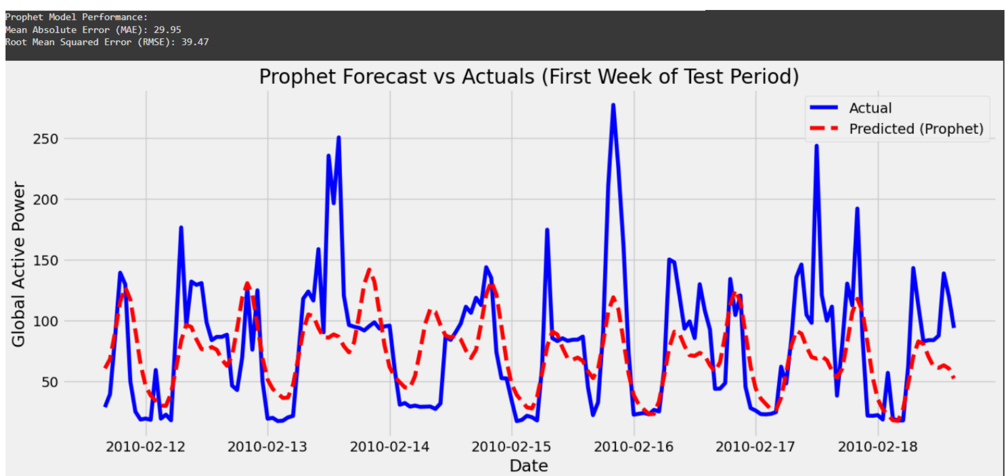

Figure 2: Prophet Forecast vs Actuals (Dataset 1)

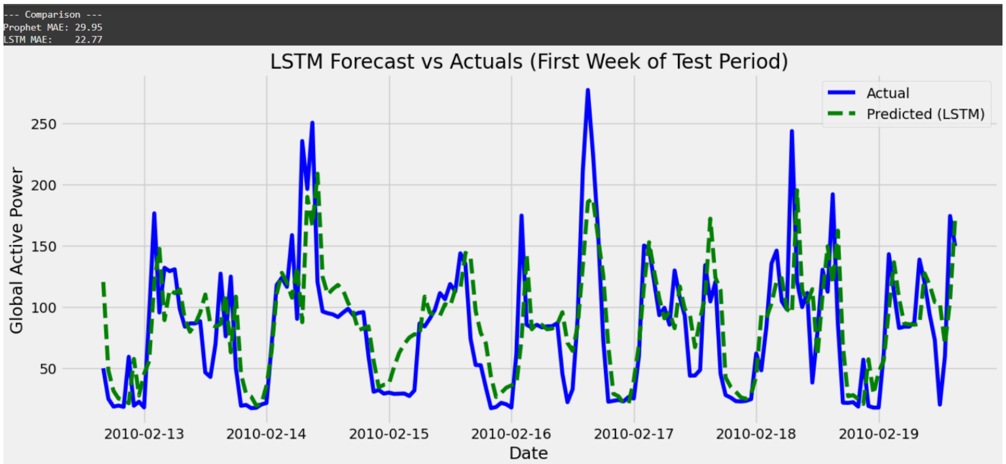

Figure 3: LSTM Forecast vs Actuals (Dataset 1)

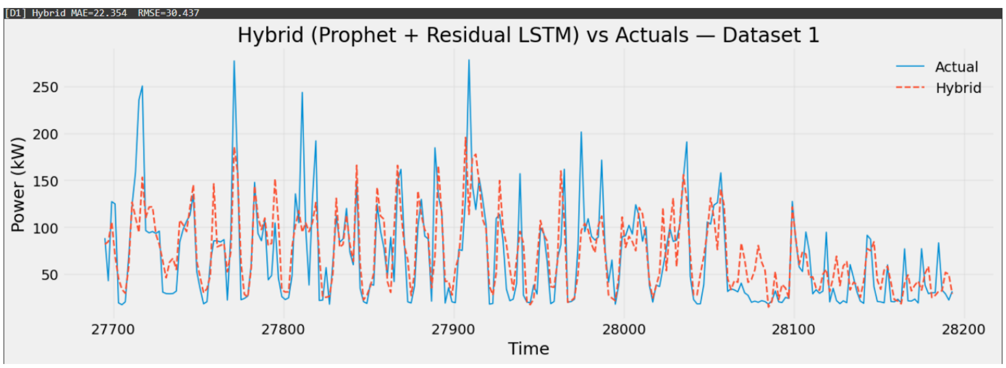

Figure 4: Hybrid Forecast vs Actuals (Dataset 1)

5.1.2 Dataset 2: Electricity Load Diagrams

The Hybrid model performed best with dataset two as well. It showed large error reductions, when compared to Prophet and better accuracy than the LSTM. This is most likely because the dataset has stron, seasonal values and sharp spikes. The Hybrid model worked well here because Prophet handled the main values, while the LSTM learned to correct the sharp spikes that Prophet missed.

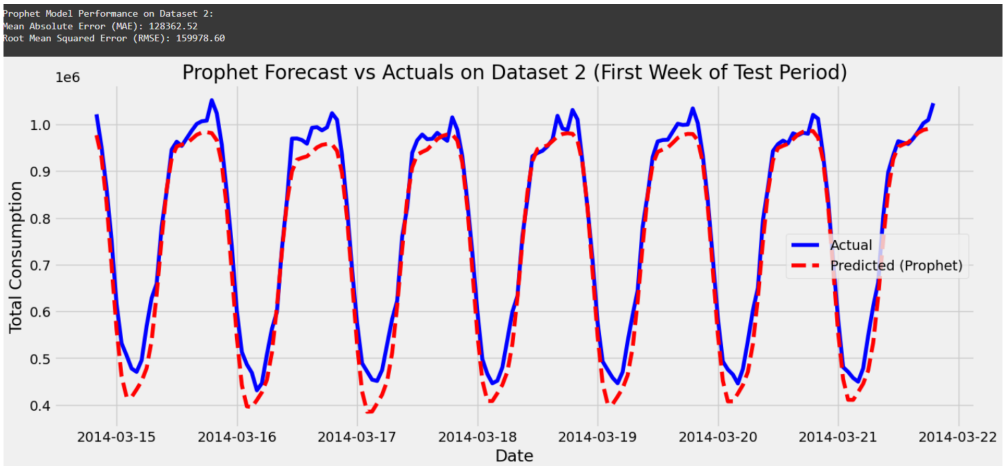

Figure 5: Prophet Forecast vs Actuals (Dataset 2)

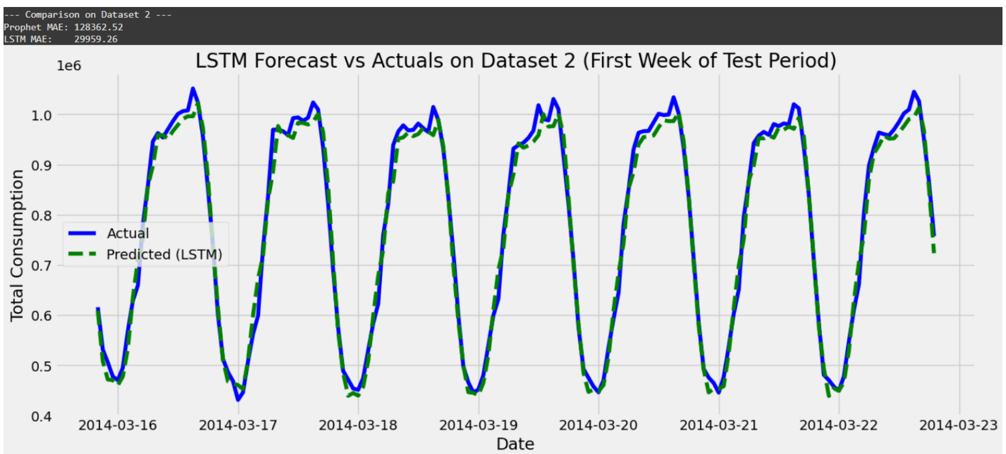

Figure 6: LSTM Forecast vs Actuals (Dataset 2)

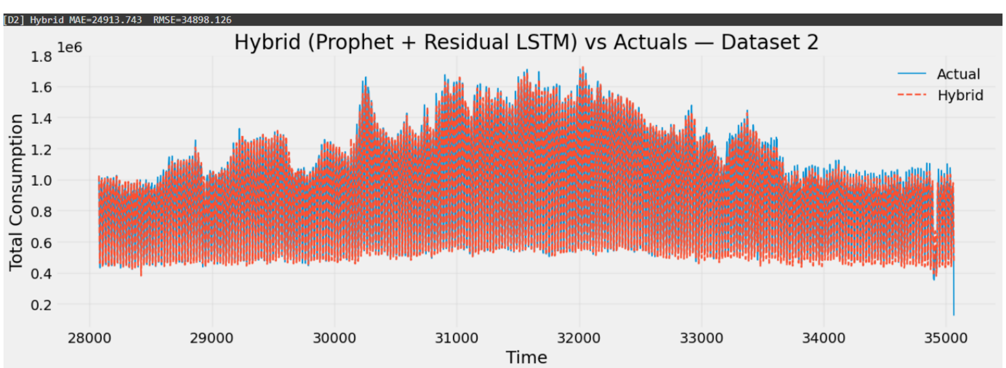

Figure 7: Hybrid Forecast vs Actuals (Dataset 2)

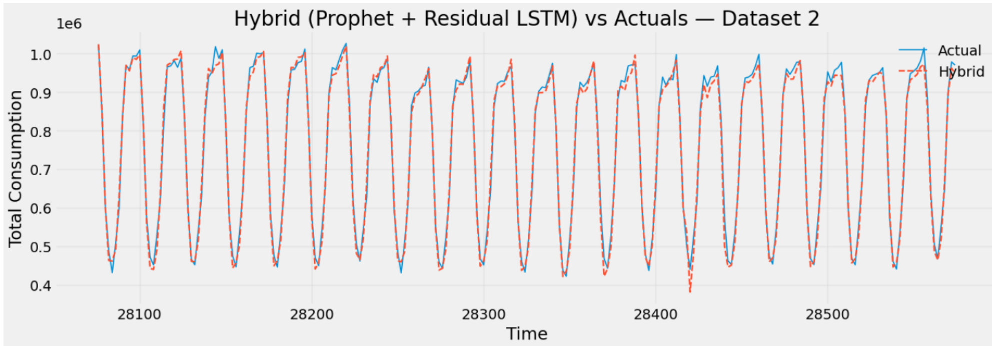

Figure 8: Hybrid Forecast vs Actuals (Dataset 2, Zoomed)

5.1.3 Dataset 3: Electricity Transformer Temperature

Electricity Transformer Temperature (ETT) (Dataset three) The LSTM outperformed the other two models. Prophet predictions were out of sync with the actual data (phase lag) and consistently missed the correct height of the temperature swings (amplitude bias). The Hybrid showed little improvement. A LSTM is already suited to catch this type of dynamic, so predicting Prophet's residuals didn't help at all. Hybrid models performed best when the data had strong seasonal values and complex spikes (Dataset one and two), while the LSTM won on the smooth data (Dataset three).

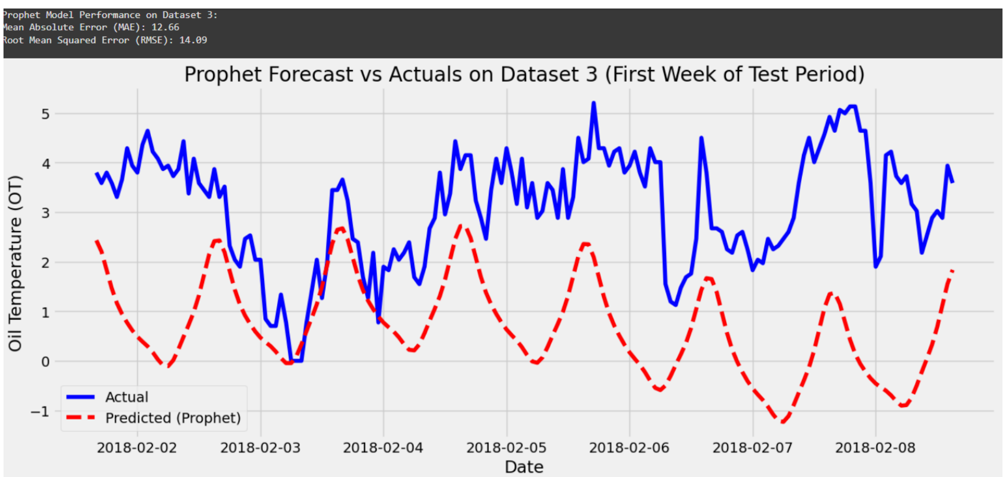

Figure 9: Prophet Forecast vs Actuals (Dataset 3)

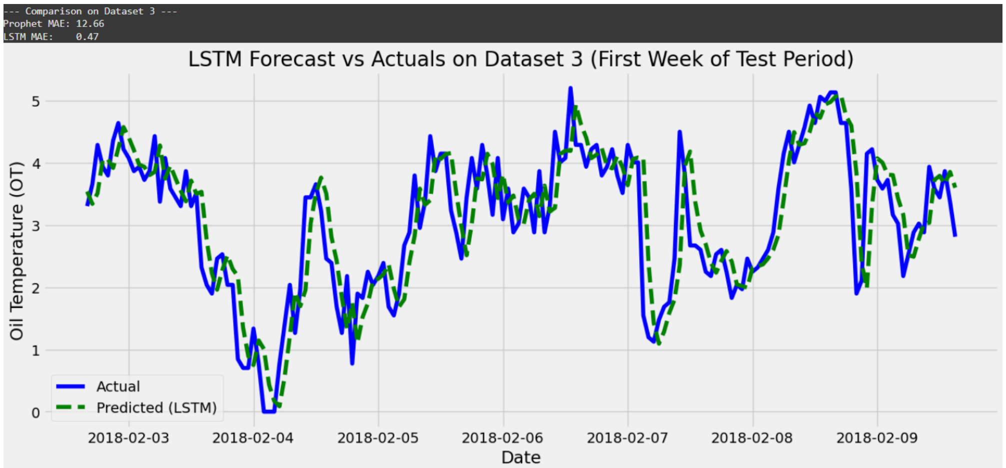

Figure 10: LSTM Forecast vs Actuals (Dataset 3)

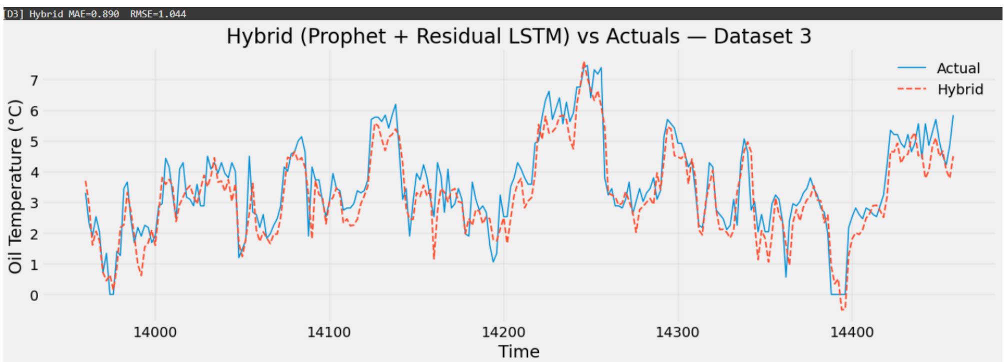

Figure 11: Hybrid Forecast vs Actuals (Dataset 3)

6 Discussion

The Hybrid model won on Dataset one and two because those datasets had predictable daily and weekly values. Prophet's good at catching basic structure to provide a stable baseline forecast. The LSTM part of the hybrid model only had to learn the residuals (ex. small spikes and errors) after that. However, this did not work on dataset three. On dataset three, the LSTM model was most accurate. Prophet's model, which tries to fit seasonal patterns, did not match the data well and introduced bias. This meant the residuals that the hybrid's LSTM had to learn were not helping, so the hybrid model benefit over the simpler, solo LSTM model. This suggests Prophet is still a good choice when it comes to management or policy decisions. However, when the only thing that matters is getting more accurate immediate predictions on more complex data, the results show that an LSTM or a Hybrid model is the better choice. [1-7]

7 Conclusion and Future Work

This study's comparison showed that applying the best performing model depended on the data's structure and characteristics. The Hybrid model performed best on seasonal and noisy data because Prophet handled main cycles and the LSTM fixed the residuals. For the smoother sensor dataset (dataset three), the LSTM model was more accurate, because Prophet did not fit well which also caused it to reduce the hybrid's performance. This reinforces the idea that the model should be chosen based on the data's specific patterns.

Future work could build on this by testing longer time periods (months, years), adding weather data, or using more advanced hybrid models/other models in general for comparison. [1-7]

References

[1] X. Yang, S. Zhao, K. Li, W. Chen, S. Zhang, and J. Chen, “An optimized short-term load forecasting method based on time and weather-fused features using a ConvLSTM-3D; Prophet used for temporal decomposition,” *Frontiers in Energy Research*, vol. 12, p. 1501963, Jan. 2025. [Online]. Available: https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2024.1501963/full

[2] T. Bashir, H. Chen, M. F. Tahir, and L. Zhu, “Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN,” *Energy Reports*, vol. 8, pp. 1678–1686, Nov. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484721015067

[3] S. Chaturvedi, P. E. Morthala, and J. Ye, “A comparative assessment of SARIMA, LSTM RNN and Fb Prophet models to forecast total and peak monthly energy demand for india,” *Energy Policy*, vol. 168, p. 113097, Sep. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0301421522003226

[4] S. Arslan, “A hybrid forecasting model using LSTM and Prophet for energy consumption with decomposition of time series data,” *PeerJ Computer Science*, vol. 8, p. e1001, Jun. 2022. [Online]. Available: https://www.researchgate.net/publication/361222023_A_hybrid_forecasting_model_using_LSTM_and_Prophet_for_energy_consumption_with_decomposition_of_time_series_data

[5] Q. Dong, R. Huang, C. Cui, D. Towey, L. Zhou, J. Tian, and J. Wang, “Short-term electricity-load forecasting by deep learning: A comprehensive survey,” *Engineering Applications of Artificial Intelligence*, vol. 154, p. 119421, Aug. 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0952197625009807

article fontspec [english, bidi=basic, provide=*]babel [import, onchar=ids fonts]english rmNoto Sans

AI Usage Disclosure

The LaTeX code within Overleaf, received assistance from generative AI models, specifically Google Gemini and ChatGPT. As I am not very familiar with coding for overleaf.