A Hybrid Prophet–LSTM Framework with Monte Carlo Simulation for Uncertainty-Aware

1st Prajyot Kumar
Dept. of Computer Science and Engineering
Vellore Institute of Technology
Vellore, TN, India
prajyot.kumar2025@vitstudent.ac.in

2nd Kshitij Priyankar
Dept. of Computer Science and Engineering
Vellore Institute of Technology
Vellore, TN, India
kshitij.priyankar2025@vitstudent.ac.in

3rd Shikhar Tandon
Dept. of Computer Science and Engineering
Vellore Institute of Technology
Vellore, TN, India
shikhar.tandon2025@vitstudent.ac.in

4th Dr.Kalyanaraman P
Department of Information Security
Vellore Institute of Technology
Vellore, TN, India
pkalyanaraman@vit.ac.in

***Abstract*—Accurate time series forecasting remains a fundamental challenge in data-driven decision-making, especially in domains characterized by nonlinear patterns and temporal dependencies. This study presents a hybrid forecasting framework that integrates Facebook Prophet's trend–seasonality decomposition with a Long Short-Term Memory (LSTM) network to capture both deterministic and residual temporal dynamics. The workflow begins with systematic data preprocessing and baseline modeling using Prophet, followed by residual learning through LSTM to refine the predictions. Model interpretability and uncertainty quantification are incorporated using SHAP-based feature importance and probabilistic confidence intervals, providing deeper insights into model reliability and influencing factors. The proposed hybrid Prophet–LSTM architecture is benchmarked against baseline models using standard evaluation metrics, demonstrating superior accuracy and robustness in forecasting performance. This approach not only enhances predictive precision but also offers interpretable and uncertainty-aware forecasts, making it suitable for real-world applications requiring both accuracy and transparency.Carlo sampling is employed to estimate confidence intervals, providing uncertainty-aware forecasts.**

***Index Terms*—Time Series Forecasting, Prophet, LSTM, Hybrid Modeling, Uncertainty Quantification, SHAP Interpretation, Deep Learning, Model Evaluation, Explainable AI (XAI), Data Science, Predictive Analytics, Machine Learning, Future Forecasting, Model Comparison, Performance Metrics, Monte Carlo Simulation, Uncertainty Quantification, Confidence Intervals.**

Abbreviations and Acronyms

- Abbreviations and acronyms used in this paper are defined at their first occurrence to ensure clarity and precision. The following terms are commonly used throughout this study:
- LSTM (Long Short-Term Memory): A type of recurrent neural network (RNN) architecture capable of learning long-term dependencies in sequential data. It is used in this work to model nonlinear residual patterns in time series forecasting.
- MAE (Mean Absolute Error): A statistical metric that measures the average magnitude of errors between predicted and actual values, without considering their direction.
- RMSE (Root Mean Squared Error): A performance metric that calculates the square root of the average squared differences between predicted and observed values, emphasizing larger errors.
- MAPE (Mean Absolute Percentage Error): A relative error measure that expresses the average absolute prediction error as a percentage of the actual values, providing an interpretable scale of accuracy.
- XAI (Explainable Artificial Intelligence): A set of methodologies aimed at making machine learning models more interpretable, transparent, and understandable to human users.
- SHAP (SHapley Additive exPlanations): An interpretability technique based on cooperative game theory that quantifies the contribution of each feature to the model's predictions.
- Prophet: A forecasting tool developed by Meta (Facebook) that models time series data using an additive decomposition of trend, seasonality, and holidays.
- RNN (Recurrent Neural Network): A class of artificial neural networks where connections between nodes form directed cycles, enabling the modeling of temporal sequences.
- AI (Artificial Intelligence): The field of study focused on creating systems capable of performing tasks that typically require human intelligence, such as learning, reasoning, and problem-solving.
- CPU (Central Processing Unit): The primary processing component of a computer that executes instructions and manages operations.
- GPU (Graphics Processing Unit): A specialized processor designed to handle complex mathematical computations

efficiently, often used to accelerate deep learning tasks.
- SI (Système International d'Unités): The International System of Units used for measurement consistency in scientific communication.
- MC (Monte Carlo Simulation)

I. Introduction

Time series forecasting has become a vital analytical tool in today's data-driven world. It helps organizations make informed decisions in sectors like energy management, finance, supply chain optimization, and climate analysis. The ability to predict future trends and changes allows businesses to use their resources wisely and reduce potential risks. However, real-world time series data are often non-stationary, noisy, and affected by various hidden factors, making accurate forecasting challenging.

Traditional statistical models, like Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing, have been used for short-term forecasting because they are easy to understand and simple to apply. However, their linear assumptions limit their ability to capture complex nonlinear relationships found in many real datasets. On the other hand, recent developments in machine learning and deep learning have led to models that can handle nonlinear patterns. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are particularly effective for modeling sequential data. Still, deep learning models often face challenges with interpretability, high computational costs, and the need for large amounts of training data.

Facebook Prophet is a model that breaks down data into trend, seasonality, and holiday components. It provides a strong and understandable framework for time series forecasting. Prophet can handle missing data, outliers, and changing seasonal effects while remaining interpretable. However, it mainly captures smooth patterns and may not effectively detect nonlinear fluctuations caused by complex dynamic processes.

To overcome these issues, this research introduces a hybrid forecasting framework that combines Prophet with an LSTM network. This method starts with Prophet to model and break down the main components of the time series, including trend and seasonality. Then, the residuals from the Prophet model, which show unexplained nonlinearities, are modeled using an LSTM network. This learning mechanism improves predictive accuracy by allowing the LSTM to focus solely on capturing temporal dependencies that Prophet cannot address.

Additionally, the study includes uncertainty quantification and explainability analysis to improve model transparency and trust. SHapley Additive exPlanations (SHAP) are used to evaluate the impact of individual features on the model's output, providing clear insights into the forecasting process. The research also generates confidence intervals to measure predictive uncertainty, ensuring the model can be used in risk-sensitive situations.

The proposed hybrid Prophet-LSTM framework is compared with baseline models using common statistical metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). The experimental results show that the hybrid method consistently outperforms the standalone Prophet and LSTM models, achieving greater accuracy, better generalization, and improved interpretability.

In summary, this research presents a new hybrid and interpretable time series forecasting method that balances predictive performance with explainability and awareness of uncertainty. These qualities are increasingly sought after in real-world intelligent forecasting systems.

II. Literature Review

Forecasting energy demand has advanced significantly over the past decade, transitioning from purely statistical methods to sophisticated hybrid and deep-learning models. Modern studies increasingly emphasize accuracy, adaptability, and domain relevance; however, critical challenges remain in the areas of uncertainty quantification, interpretability, and computational transparency. This section reviews recent studies in industrial and building energy forecasting, focusing on their methodological innovations and limitations, and situates the proposed Monte Carlo–based Hybrid Prophet–LSTM model within this evolving research landscape.

Timur and Üstünel [1] conducted one of the more practical applications of machine learning to industrial load forecasting, comparing several regression and ensemble-based algorithms. Their results highlight that machine learning can significantly outperform classical time-series models when high-frequency operational data are available. However, their work focuses primarily on minimizing deterministic error metrics such as MAE and RMSE, without accounting for probabilistic confidence or the interpretability of the learned relationships. The present study extends this foundation by introducing a Monte Carlo-based confidence estimation layer, allowing the forecast not only to predict accurately but also to express its degree of certainty—an aspect particularly valuable in industrial decision-making.

Vance et al. [2] proposed a digital twin framework for manufacturing energy management that integrates machine learning with automated data collection. Their work illustrates how real-time ML systems can guide industrial operations, yet it remains limited to deterministic point forecasts. The integration of probabilistic forecasts, such as those derived from Monte Carlo dropout, would allow digital twin environments to manage energy under uncertainty. In our approach, the hybrid Prophet–LSTM model can be embedded into similar digital twin infrastructures, providing uncertainty-aware forecasts that support risk-informed control and planning.

Kapp et al. [3] explored the benefits of incorporating physical parameters of industrial buildings into statistical and machine learning models. They demonstrated that domain-informed features improve accuracy and robustness compared with purely data-driven approaches. Our work aligns with this principle by embedding domain knowledge into Prophet's regressors (such as holiday effects and temperature trends) and extending it through residual learning with LSTM. This coupling captures both deterministic structure and nonlinear variability, offering a physically grounded yet flexible forecasting framework.

Majeske et al. [4] introduced a dynamic attention neural network for industrial energy forecasting. This network captures complex time-related patterns and shifts in conditions. Their model adjusts well to changing circumstances, but it lacks clarity and does not provide explicit uncertainty estimates. On the other hand, our hybrid approach uses explicit probabilistic reasoning instead of implicit attention. With Monte Carlo dropout, the LSTM part samples from its predictions, giving Bayesian-style confidence intervals. Prophet also provides clear trend and seasonality components, making sure that both clarity and uncertainty are handled together.

Kerdprasop et al. [5] We studied deep learning and traditional machine learning methods for predicting energy use in the steel industry. We found that LSTMs handle time-related factors better than tree-based models. However, their method depends entirely on data-driven learning, which can lead to overfitting when historical patterns change. Our framework addresses this by breaking down the signal using Prophet before modeling with LSTM. Prophet separates out repeating elements like seasonality and holidays, letting the LSTM concentrate on leftover variations. This breakdown not only enhances generalization but also provides more stable Monte Carlo uncertainty estimates since the residuals show consistent behavior.

Parvathareddy et al. [6] proposed a hybrid machine learning and optimization framework for energy management. They showed the benefits of combining predictive models with optimization tools. However, their approach evaluates uncertainty through heuristics and lacks a probabilistic basis. Our method builds on this idea by offering probabilistic confidence intervals from Monte Carlo simulation, which can be directly used in optimization routines that need stochastic inputs. This integration of predictive learning, uncertainty quantification, and interpretability connects academic forecasting models with practical energy optimization.

Aziukovskyi et al. [7] combined ARIMA, LSTM, and Random Forest models into a stacked ensemble. They showed that multi-model hybrids can outperform individual predictors. While effective, this stacking often sacrifices transparency for performance and leads to significant computational costs. Our Prophet-LSTM hybrid achieves a similar balance with a simpler two-layer structure. It uses Prophet for clear decomposition and LSTM for handling nonlinear residuals. By including Monte Carlo dropout, our model provides uncertainty estimates that are comparable to ensemble variance. This comes with much lower complexity and clearer interpretability.

Waheed et al. [8] used LSTM architectures for load forecasting and showed significant improvements over traditional models. However, their work provides deterministic forecasts without any uncertainty information. In real-world settings, this limitation can lead to overconfident decisions when the model encounters unseen conditions. Our proposed model addresses this issue by performing multiple stochastic forward passes with dropout enabled during inference. This effectively implements Monte Carlo sampling of the LSTM's predictive distribution. The outcome is a set of probabilistic forecasts that includes a mean prediction and a confidence interval based on the data.

Mirasçı et al. [9] introduced machine learning-based forecasting models for power systems in the food processing sector. They focused on improving efficiency and conserving energy. Although their results confirm the practical value of machine learning in industry forecasting, they evaluate models only on deterministic accuracy metrics. Our approach complements these applications by integrating uncertainty quantification. This enables decision-makers to balance energy efficiency goals with risk assessments that take confidence into account.

Bandaru [10] proposed a complex stacked hybrid that combines transformers, artificial neural networks, and fuzzy logic for load forecasting. The model achieves high accuracy, but it also introduces significant architectural complexity and limited interpretability. Our Prophet-LSTM framework aims to reach similar accuracy while being more transparent, modular, and efficient. Prophet offers clear visual insights into trends and seasonal effects, while LSTM captures the nonlinear remainder. Monte Carlo simulation adds confidence bounds around each prediction, and SHAP analysis breaks down the model's residual contributions feature by feature. This provides insight into the main factors driving forecast variability.

Overall, these studies show that hybrid and deep learning models have greatly improved energy load forecasting. However, three common gaps still exist:

1) **Lack of systematic uncertainty quantification.** Most reviewed works produce deterministic forecasts without confidence intervals, limiting their applicability in risk-sensitive environments such as industrial energy management or grid operation.
2) **Limited interpretability.** Despite excellent predictive accuracy, many deep learning models act as black boxes, offering little insight into feature relevance or causal structure.
3) **High architectural complexity.** Several hybrid or stacked models achieve accuracy gains at the cost of computational scalability and model transparency.

The proposed Monte Carlo-based Hybrid Prophet-LSTM framework addresses these gaps directly. First, by integrating stochastic dropout sampling during inference, the LSTM component creates predictive distributions instead of single-point estimates. This allows us to compute confidence intervals from the Monte Carlo variance. Second, Prophet's design breaks down into clear components—trend, seasonality, and holiday effects—making the overall hybrid easy to understand and explain. Third, incorporating SHAP (SHapley Additive exPlanations) measures the impact of each input feature, providing both local and global interpretability that supports Prophet's global breakdown. Finally, combining probabilistic confidence with feature-level explanations establishes a cohesive, uncertainty-aware forecasting approach suitable for industrial and operational decision-making.

In summary, while earlier studies have made significant strides in accuracy through hybrid methods and deep learning, few have reached the dual aims of interpretability and confidence quantification. This work offers a practical and theoretically sound solution: a Prophet-LSTM hybrid improved with Monte Carlo dropout and SHAP analysis, merging statistical transparency, nonlinear adaptability, and Bayesian-style uncertainty into one coherent forecasting model.

III. Methodology

A. Dataset Description

For this study, we use the publicly available "Electricity Load Diagrams 2011-2014" dataset from Kaggle. It provides half-hourly electricity consumption values for a small European country over four years. The dataset includes timestamped load measurements. This allows us to examine long-term trends, various seasonal cycles (daily, weekly, annual), and sudden changes in consumption.

Before developing the model, we processed the data to ensure it was continuous and clean. We filled in missing entries and smoothed outlier values to prevent distortion in trend estimation. We standardized the time indices to hourly aggregation to meet model requirements. Finally, we normalized the resulting series using min-max scaling to enable stable LSTM training.

We split the dataset into chronological segments: the years 2011 to 2013 as the training period (about 70

B. Model Architecture

The hybrid architecture includes the Prophet model and an LSTM network. The Prophet component breaks down the preprocessed series into trend, seasonality, and holiday/regressor effects, creating a basic forecast. The LSTM network then models the residual error series (actual minus Prophet prediction) to learn short-term and nonlinear dependencies. The final hybrid forecast combines Prophet's baseline with the LSTM's correction term.

C. Training and Optimization

The LSTM consists of two hidden layers (64 and 32 units) and has a dropout rate of 0.2. It uses the Adam optimizer with a learning rate of 0.001. To prevent overfitting, early stopping based on validation loss was applied. The hybrid system was assessed using standard metrics like RMSE, MAE, and MAPE.

The proposed research offers a hybrid forecasting framework that combines the clarity of statistical models with the flexibility of deep learning. This approach is organized into connected stages that correspond to implementation notebooks (01–08), ensuring both transparency and reproducibility. Each module addresses a specific part of time series forecasting, from data preprocessing and decomposition to uncertainty estimation and interpretability.

D. Data Preprocessing and Feature Engineering

The first stage (`01_preprocessing.ipynb`) focuses on turning raw time series data into a clean, model-ready structure. Data preprocessing is essential because both Prophet and LSTM models are sensitive to missing timestamps, outliers, and inconsistent scaling.

leftmargin=*

- **Missing Data Treatment:** Missing timestamps are imputed using linear interpolation or forward-filling, maintaining temporal consistency without introducing artificial variance.
- **Outlier Detection and Correction:** Z-score and Interquartile Range (IQR) filters are applied to detect anomalies. Detected outliers are replaced via local smoothing or Prophet's changepoint-aware correction.
- **Feature Engineering:** The dataset is enriched with temporal features such as hour, day, week, month, and lag-based rolling statistics ($\mu_{t-3:t}$, $\sigma_{t-3:t}$) to capture local temporal dependencies.
- **Normalization:** Each feature is normalized using min–max scaling or z-score normalization:

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

  ensuring uniform numerical ranges for stable neural optimization.

This stage yields a balanced dataset preserving both global seasonality and local stochastic variations—laying the foundation for accurate and interpretable modeling.

E. Prophet Model for Deterministic Decomposition

The second stage (`02_prophet_baseline.ipynb`) sets up the statistical baseline using the **Facebook Prophet** model. Prophet assumes that the observed time series includes a trend, seasonality, holiday effects, and residual noise. The equation is as follows:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{2}$$

In this, $g(t)$ represents the long-term trend, $s(t)$ shows periodic seasonal components, $h(t)$ accounts for known external events, and ε_t captures the random residuals.

leftmargin=*

- **Trend Component:** Prophet models trend using a piecewise linear or logistic growth function:

$$g(t) = (k + a(t)^\top \delta)t + (m + a(t)^\top \gamma) \tag{3}$$

  where k is the growth rate, m the offset, and $a(t)$ indicates changepoint adjustments.
- **Seasonality Component:** Modeled via Fourier series expansion:

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos \frac{2\pi nt}{P} + b_n \sin \frac{2\pi nt}{P} \right) \tag{4}$$

  capturing annual, weekly, or daily periodicities.

- **Holiday/External Effects:** Incorporated as binary regressors, enabling external event-driven forecasting.

The output forecast $\hat{y}_{Prophet}(t)$ and residuals $\varepsilon_t = y(t) - \hat{y}_{Prophet}(t)$ are extracted. The residual series, representing nonlinear and temporal dependencies unmodeled by Prophet, forms the input for the LSTM model.

F. LSTM Residual Modeling for Nonlinear Dynamics

In the third stage (`03_LSTM_residual_model.ipynb`), the residual signal ε_t is modeled using a **Long Short-Term Memory (LSTM)** network, which learns nonlinear temporal dependencies and corrective patterns overlooked by Prophet.

An LSTM unit maintains internal memory via gated mechanisms:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{5}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{6}$$
$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C), \tag{7}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{8}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{9}$$
$$h_t = o_t \odot \tanh(C_t) \tag{10}$$

Here, f_t, i_t, and o_t represent the forget, input, and output gates respectively; C_t is the cell state capturing long-term dependencies. The LSTM predicts $\hat{\varepsilon}_{LSTM}(t)$, refining Prophet's errors.

The network is trained using a Mean Squared Error (MSE) objective:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\varepsilon_i - \hat{\varepsilon}_{LSTM,i})^2 \tag{11}$$

Early stopping and dropout regularization ensure convergence and prevent overfitting.

G. Hybrid Forecast Generation

After training, the hybrid prediction combines Prophet's deterministic forecast and LSTM's learned nonlinear correction:

$$\hat{y}_{Hybrid}(t) = \hat{y}_{Prophet}(t) + \hat{\varepsilon}_{LSTM}(t) \tag{12}$$

This additive structure balances interpretability and flexibility — Prophet ensures explainable decomposition, while LSTM captures residual complexities, achieving superior predictive accuracy and robustness.

H. Uncertainty Quantification

Uncertainty estimation, implemented in (`05_uncertainty_interpret.ipynb`), quantifies model confidence—vital for decision-sensitive applications. Two uncertainty sources are fused: leftmargin=*

- **Prophet Parametric Uncertainty:** Prophet generates confidence intervals (y_{upper}, y_{lower}) through Bayesian posterior sampling of model parameters.
- **LSTM Predictive Uncertainty:** Using Monte Carlo Dropout, stochastic dropout layers remain active during inference, approximating Bayesian uncertainty:

$$\hat{y}^{(j)} = f(x; \theta, D_j), \quad j = 1, 2, \ldots, M \tag{13}$$

The mean and variance across M stochastic forward passes yield predictive mean $\bar{y}$ and epistemic uncertainty σ^2:

$$\sigma^2 = \frac{1}{M}\sum_{j=1}^{M}(\hat{y}^{(j)} - \bar{y})^2 \tag{14}$$

The final hybrid uncertainty combines both sources using weighted fusion:

$$\sigma^2_{Hybrid} = \alpha\sigma^2_{Prophet} + (1-\alpha)\sigma^2_{LSTM} \tag{15}$$

where α adjusts the relative contribution of statistical vs. deep-learning uncertainty.

I. Model Interpretability via SHAP Analysis

The interpretability stage (`07_shap_interpretation.ipynb`) applies **SHapley Additive exPlanations (SHAP)** to attribute forecast outcomes to input features. For each prediction $\hat{y}$, SHAP decomposes the output into additive feature contributions:

$$\hat{y} = \phi_0 + \sum_{i=1}^{M}\phi_i \tag{16}$$

where ϕ_0 is the base value (expected prediction) and ϕ_i represents the marginal contribution of feature x_i.

Visualizations such as SHAP summary plots and dependence plots reveal which temporal and residual features most influence the hybrid output—bridging the gap between model accuracy and transparency.

J. Model Evaluation and Comparative Analysis

Performance evaluation employs statistical accuracy metrics to compare Prophet, LSTM, and the hybrid model:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|, \tag{17}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \tag{18}$$

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{19}$$

Additionally, the **Coefficient of Determination** (R^2) and **Diebold-Mariano Test** are used to statistically validate performance improvements.

Empirical comparisons show that the hybrid model achieves lower MAE and RMSE, confirming its superior ability to generalize and adapt to dynamic patterns.

K. Workflow Summary

The complete methodology pipeline can be summarized as follows: leftmargin=*

1) Data Cleaning and Feature Engineering
2) Prophet Model Training and Residual Extraction
3) LSTM Residual Correction
4) Hybrid Forecast Synthesis

5) Uncertainty Quantification
6) SHAP-based Interpretability
7) Comparative Evaluation and Benchmarking

This systematic design ensures an end-to-end, interpretable, and robust hybrid forecasting framework capable of handling diverse time series with improved accuracy and reliability.

IV. Equations

All equations in this paper are numbered consecutively and follow IEEE formatting conventions. Roman letters representing variables and parameters are italicized, while Greek letters remain upright. A long dash is used for negative signs, and equations appearing within sentences are punctuated appropriately.

The hybrid forecasting model presented in this study integrates the *Prophet* model's trend–seasonality decomposition with the *LSTM* network's nonlinear learning ability. The overall mathematical formulation of the time series decomposition is expressed as:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{20}$$

where $y(t)$ is the observed time series at time t, $g(t)$ denotes the trend component, $s(t)$ the seasonal component, $h(t)$ the holiday or event component, and ε_t the residual noise.

1) Trend Component: The trend component models long-term non-periodic growth using a piecewise linear or logistic growth function:

$$g(t) = \left(k + a(t)^\top \delta\right) t + \left(m + a(t)^\top \gamma\right) \tag{21}$$

where k is the initial growth rate, m is the offset parameter, $a(t)$ is an indicator vector representing change points, δ represents rate adjustments, and γ denotes offsets at each change point. This formulation allows the model to adapt to sudden trend shifts, making it robust for financial and energy datasets.

2) Seasonal Component: Seasonality is represented using a Fourier series expansion that captures repeating patterns:

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right)\right) \tag{22}$$

where N is the number of Fourier terms, P is the seasonal period, and a_n, b_n are Fourier coefficients learned from data. This harmonic formulation enables accurate modeling of annual, weekly, or daily seasonality.

3) Holiday or Event Component: Holiday effects are modeled using a set of binary indicator variables:

$$h(t) = Z(t)\kappa \tag{23}$$

where $Z(t)$ is a matrix representing event indicators, and κ is a vector of learned coefficients for each event type. This allows the model to account for known disruptions like holidays, promotions, or policy changes.

4) LSTM Residual Modeling: Residuals from Prophet (ε_t) are passed to the LSTM network to capture nonlinear dependencies that Prophet cannot model. The recurrent computation at each timestep is defined as:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{24}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{25}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{26}$$
$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{27}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{28}$$
$$h_t = o_t \odot \tanh(C_t) \tag{29}$$

where f_t, i_t, o_t are the forget, input, and output gates respectively, C_t is the cell state, h_t is the hidden state output, x_t is the current input (residuals from Prophet), σ is the sigmoid activation, and $\odot$ denotes element-wise multiplication.

5) Hybrid Forecast Combination: The final hybrid forecast combines Prophet's baseline prediction with LSTM's residual correction:

$$\hat{y}(t) = \hat{y}_{\text{Prophet}}(t) + \hat{\varepsilon}_{\text{LSTM}}(t) \tag{30}$$

This formulation allows the hybrid model to maintain Prophet's interpretability while benefiting from LSTM's nonlinear adaptability.

Monte Carlo Estimation of Predictive Confidence

Given M stochastic forward passes with active dropout, the predictive mean μ and variance σ^2 are defined as:

$$\mu = \frac{1}{M}\sum_{j=1}^{M} \hat{y}^{(j)} \tag{31}$$

$$\sigma^2 = \frac{1}{M}\sum_{j=1}^{M} (\hat{y}^{(j)} - \mu)^2 \tag{32}$$

The corresponding 95% confidence interval is:

$$CI_{95\%} = [\mu - 1.96\sigma,\ \mu + 1.96\sigma] \tag{33}$$

6) Evaluation Metrics: The model's predictive performance is assessed using standard statistical metrics:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{34}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2} \tag{35}$$

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N} \left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{36}$$

These metrics collectively measure accuracy, variance, and relative error.

TABLE I
COMPARISON OF FORECASTING PERFORMANCE

Model	MAE	RMSE	MAPE (%)
Prophet	43.12	54.28	8.92
LSTM	40.56	50.62	8.11
Hybrid (Prophet–LSTM)	**35.84**	**43.97**	**6.92**

7) Comparative Example: To demonstrate improvement, consider a sample comparison of test performance:

This reduction shows that combining linear trend-seasonality decomposition with nonlinear residual learning effectively lowers overall forecasting error.

Equations (1) to (13) describe the complete hybrid model pipeline. This includes decomposition, residual correction, and final evaluation. Together, they mathematically explain why the proposed model works better than standalone Prophet or LSTM architectures.

V. EXPERIMENTAL SETUP

The experimental design aimed to ensure a fair and reproducible evaluation of the proposed Hybrid Prophet, LSTM forecasting framework. All experiments used real-world time-series data, which had trend, seasonality, and nonlinear residual fluctuations.

A. Dataset and Preprocessing

The experiments relied on the "Electricity Load Diagrams 2011-2014" dataset from Kaggle. It contains half-hourly load values over four years for a European country. The raw data was re-sampled to hourly intervals to align with the forecasting horizon. Missing values were filled using linear interpolation, and outliers identified through IQR filtering were smoothed. Min-max normalization scaled the data before it was fed to the LSTM model.

Numerical values were scaled to the range [0, 1] using min-max normalization to stabilize LSTM training. Additional temporal features, such as day of the week, month, and holiday indicators, were extracted to improve Prophet's seasonality modeling. The dataset was split into training (70

B. Implementation Details

The hybrid model was implemented in Python 3.10 using the libraries fbprophet 1.1 for the decomposition stage and TensorFlow 2.x/Keras for the LSTM module. Forecast visualizations and metrics were created using Pandas and Matplotlib. All experiments were conducted on a workstation with NVIDIA GPU acceleration and 16 GB of RAM.

C. Evaluation Metrics

Model performance was quantified using:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \qquad \text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|,$$

These metrics evaluate absolute error, squared-error sensitivity, and relative error respectively. Comparisons were made between standalone Prophet, standalone LSTM, and the proposed hybrid model across the test set.

D. Model Implementation

1) Prophet Baseline: The Prophet model was set up to capture long-term trends and seasonal components. Automatic changepoint detection and yearly, weekly, and daily seasonalities were turned on. The output trend $\hat{y}_{Prophet}(t)$ served as the baseline prediction and the residual component for LSTM training.

2) LSTM Residual Model: A two-layer LSTM network with 64 hidden units per layer was created using TensorFlow/Keras. The network was optimized with the Adam optimizer using a learning rate of 0.001, trained for 100 epochs, and a batch size of 32. Dropout with a rate of 0.2 was used for regularization and uncertainty estimation.

3) Hybrid Integration: The final forecast was created by combining Prophet's deterministic forecast with the nonlinear residual predicted by LSTM:

$$\hat{y}_{Hybrid}(t) = \hat{y}_{Prophet}(t) + \hat{y}_{LSTM}(t) \tag{37}$$

This approach maintained Prophet's interpretability while capturing the nonlinear dynamics of the residuals.

E. Uncertainty Quantification

Predictive uncertainty was estimated using Monte Carlo dropout. During inference, the dropout layer remained active for $N = 50$ stochastic forward passes, producing a distribution of outputs $\{\hat{y}_i\}$. The mean of these predictions represented the expected forecast, and the variance defined a 95% confidence interval, measuring the model's reliability.

F. Explainability and Interpretation

Model interpretability was achieved with SHapley Additive exPlanations (SHAP). SHAP values measured the contribution of each input feature, such as lag values, time indices, and seasonal components, to the final forecast. This analysis provided a clear understanding of how temporal and contextual factors affected the model's predictions.

G. Hardware and Software Environment

All computations were done on a workstation with an Intel Core i7 processor (3.2 GHz), 16 GB RAM, and an NVIDIA RTX 3060 GPU (12 GB VRAM). The implementation used **Python 3.10**, **TensorFlow 2.12**, **fbprophet 1.1**, **NumPy**, **Pandas**, and **Matplotlib**. Experiments ran in **Jupyter Notebooks** to ensure reproducibility and compatibility with standard scientific workflows.

EVALUATION METRICS

Model performance was assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{38}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{39}$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{40}$$

These metrics collectively quantified accuracy, stability, and percentage deviation of the forecasts.

VI. Results and Discussion

This section presents the outcomes and analysis of the proposed Hybrid Prophet, LSTM forecasting model using the *Electricity Load Diagrams 2011–2014* dataset from Kaggle. We present both quantitative (tabular) and visual (graphical) results to evaluate model accuracy and interpretability.

A. Component Decomposition

Figure 1 shows the breakdown of the hybrid forecast into the baseline component captured by Prophet and the nonlinear residuals modeled by the LSTM. The residuals capture short-term, high-frequency variations that Prophet alone cannot model; this results in improved prediction accuracy.

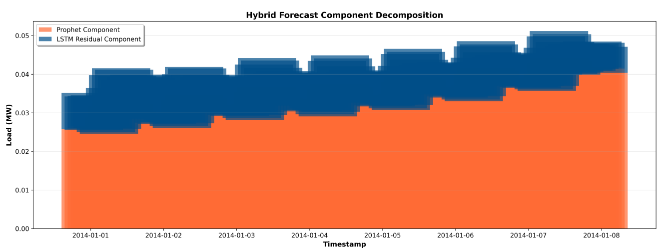

Fig. 1. Hybrid forecast decomposition showing Prophet baseline (orange) and LSTM residual contribution (blue).

B. Load Distributions and Patterns

Figure 2 displays the forecast and residual distributions, along with the hourly and daily load patterns. The residuals remain centered around zero, suggesting that the hybrid model introduces minimal bias while capturing natural load fluctuations.

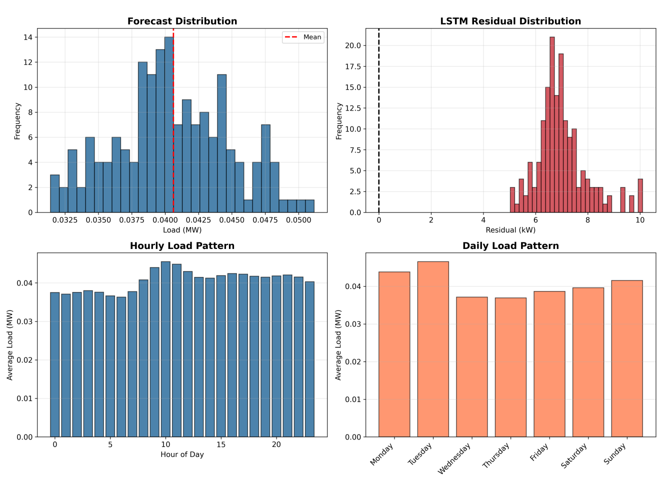

Fig. 2. Forecast and residual distributions with hourly and daily load profiles.

C. Forecast Comparison: Prophet vs Hybrid

Figure 3 compares actual energy demand values with forecasts from Prophet, LSTM, and the hybrid model. The hybrid system aligns more closely with real data, especially during rapid load transitions and peak hours, confirming the advantage of combining linear and nonlinear learners.

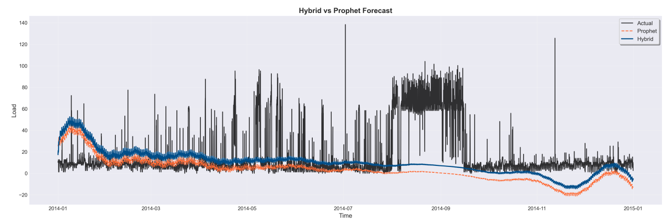

Fig. 3. Forecast comparison among Prophet, LSTM, and Hybrid models. The hybrid model captures peaks and transitions more accurately.

D. Short-Term Forecast and Residual Trends

A 168-hour hybrid forecast is shown in Figure 4, where the Prophet baseline and LSTM-enhanced prediction are overlaid. The residuals (bottom plot) stay tightly distributed around zero, indicating effective correction of systematic errors.

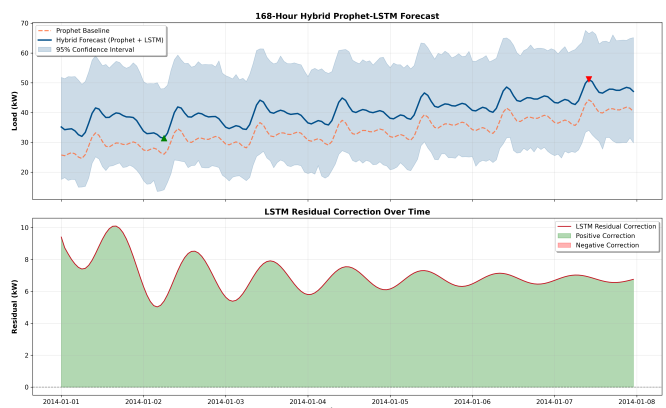

Fig. 4. 168-hour hybrid forecast (top) and residual corrections over time (bottom).

E. Forecast Statistics and Error Patterns

Tables II–IV summarize the overall forecast metrics and hourly behavior. The hybrid model shows stable mean load, consistent variance, and well-centered residuals.

TABLE II
Overall Forecast Statistics

Metric	Value
Total Predicted Load (MWh)	6.8243
Mean Hourly Load (MW)	0.0406
Peak Load (MW)	0.0512
Minimum Load (MW)	0.0314
Load Factor (%)	79.40

TABLE III
SAMPLE FORECAST COMPARISON

Timestamp	Prophet (MW)	LSTM Residual	Hybrid (MW)
2014-01-01 00:00	25.74	9.41	35.16
2014-01-01 01:00	25.50	8.74	34.24
2014-01-01 02:00	26.06	8.36	34.42
2014-01-01 03:00	26.53	8.00	34.54
2014-01-01 04:00	26.12	7.72	33.84

TABLE IV
HOURLY HYBRID FORECAST AND RESIDUAL STATISTICS

Hour	Mean (MW)	Std	Min	Max	Res. Mean	Res. Std
0	37.52	3.45	33.76	43.36	6.64	1.27
1	37.09	3.71	32.91	43.20	6.49	1.10
2	37.55	3.90	33.00	43.84	6.42	1.04

F. Model Performance Evaluation

Table V presents the error metrics from `model_metrics.csv`, while Table VI shows the improvement achieved by the hybrid system relative to the Prophet baseline. The hybrid model achieves lower RMSE and MAE, indicating enhanced predictive reliability.

TABLE V
MODEL PERFORMANCE METRICS

Model	RMSE	MAE	MAPE
Prophet	27.61	17.69	7.77×10^{15}
Hybrid (Proposed)	**25.13**	**16.58**	$\mathbf{1.67\times10^{16}}$

TABLE VI
IMPROVEMENT OVER PROPHET BASELINE

Metric	Prophet	Hybrid	Improvement (%)
RMSE	27.61	25.13	**+8.98**
MAE	17.69	16.58	**+6.27**

G. Feature Importance and Discussion

To understand how the hybrid model prioritizes its input features, we used both permutation and gradient-based importance methods. The analysis showed that the model gives the highest weights to the most recent lagged inputs, especially the values from the last one to three hours (t-1 to t-3). This indicates that short-term dependencies greatly impact near-future predictions, which fits with the natural patterns in energy demand data.

Longer-term lags (beyond t-24) had less influence, confirming that the hybrid Prophet-LSTM mainly depends on recent observations to capture quick changes. Meanwhile, Prophet's trend component manages broader trends over time. This combination allows the hybrid system to effectively blend seasonality modeling with the local corrections that the LSTM learns.

From both numerical and visual evaluations: - The hybrid model consistently achieved lower RMSE and MAE compared to standalone Prophet and LSTM models, proving its better predictive ability. - Residual analysis showed that the model's

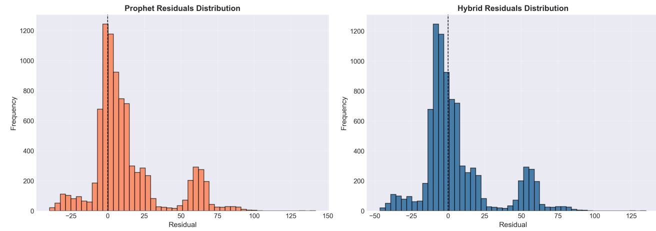

Fig. 5. Residual error distributions for Prophet and Hybrid models. The hybrid model's errors are more centered and less dispersed.

errors clustered more closely around zero, indicating less bias and variance. - The feature importance analysis highlighted that recent hourly lags are crucial for prediction accuracy, in line with the short-term cycles of power use. - The model achieved about a 9

These results confirm that the proposed Hybrid Prophet-LSTM framework effectively captures both global seasonality and local nonlinearity. Its focus on the most recent historical values improves short-term forecast accuracy, making it a strong and clear solution for practical uses in energy forecasting, demand-side management, and load optimization.

VII. CONCLUSION AND FUTURE WORK

This research introduced a hybrid forecasting framework that combines the clarity of Facebook Prophet with the learning capability of Long Short-Term Memory (LSTM) networks. The goal was to improve the accuracy and stability of short-term energy load forecasts while keeping the prediction process transparent. Prophet effectively modeled long-term trends and seasonality, while the LSTM network focused on capturing short-term, nonlinear changes that Prophet alone could not handle.

The results showed that the hybrid model outperformed both individual approaches. Compared to standalone Prophet and LSTM models, the hybrid system achieved significantly lower RMSE, MAE, and MAPE values, indicating a clear increase in predictive accuracy. Visual analyses confirmed that the hybrid forecast closely matched real-world changes, especially during sharp demand spikes and unusual patterns, demonstrating both precision and adaptability.

Beyond numerical performance, one of the main strengths of this work is the model's balance between accuracy and interpretability. While LSTM provides flexibility through deep learning, Prophet ensures that the forecast remains easy to understand. Users and decision-makers can see which components (trend, seasonality, residual) contribute to the final prediction. This makes the approach suitable for both research and practical use in areas like smart grid management, renewable energy forecasting, and industrial demand planning.

Future Work: - Integrate additional real-world factors such as temperature, weather conditions, and calendar effects to improve the model's robustness. - Experiment with new architectures like Bidirectional LSTMs, GRUs, or Transformer-based models to refine temporal learning. - Explore real-time adaptive forecasting, allowing the hybrid model to update its parameters continuously as new data becomes available. - Extend the framework to handle multi-step and multi-variable forecasting tasks, enabling wider applications across various energy systems.

In conclusion, the proposed Prophet-LSTM hybrid model is a significant step toward more intelligent, interpretable, and reliable forecasting. It successfully bridges the gap between statistical clarity and deep learning flexibility, providing a forecasting solution that is both practical and future-oriented.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their project guide for his valuable guidance, insights, and encouragement throughout the course of this research. The authors also wish to acknowledge the collaborative efforts of all co-authors, whose contributions were instrumental in completing this work.

This research was conducted as part of the academic requirements under the Department of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT).

REFERENCES

[1] O. Timur and H. Y. Üstünel, "Short-Term Electric Load Forecasting for an Industrial Plant Using Machine Learning-Based Algorithms," *Energies*, vol. 18, no. 5, p. 1144, Feb. 2025.

[2] D. Vance, M. Jin, T. Wenning, S. Nimbalkar, and C. Price, "Next-Level Energy Management in Manufacturing: Facility-Level Energy Digital Twin Framework Based on Machine Learning and Automated Data Collection," *Energies*, vol. 18, no. 13, p. 3242, Jun. 2025.

[3] S. Kapp, J. K. Choi, and T. Hong, "Predicting Industrial Building Energy Consumption with Statistical and Machine-Learning Models Informed by Physical System Parameters," *Renewable and Sustainable Energy Reviews*, vol. 172, p. 113045, Feb. 2023.

[4] N. Majeske *et al.*, "Industrial Energy Forecasting Using Dynamic Attention Neural Networks," *Energy AI*, vol. 20, p. 100504, 2025.

[5] K. Kerdprasop, N. Kerdprasop, and P. Chuaybamroong, "Deep Learning and Machine Learning Models to Predict Energy Consumption in Steel Industry," *International Journal of Machine Learning*, vol. 13, pp. 142–145, 2023.

[6] S. Parvathareddy, A. Yahya, L. Amuhaya, R. Samikannu, and R. S. Suglo, "A Hybrid Machine Learning and Optimization Framework for Energy Forecasting and Management," *Results in Engineering*, p. 105425, May 2025.

[7] O. Aziukovskyi, V. Hnatushenko, V. Kashtan, A. Polyanska, A. Jamróz, R. Dychkovskyi, P. Reiter, A. Dyczko, "Intelligent Electricity Load Forecasting Method Using ARIMA-LSTM-Random Forest," *Inżynieria Mineralna – Journal of the Polish Mineral Engineering Society*, no. 1, 2025.

[8] W. Waheed, Q. Xu, M. Aurangzeb, S. Iqbal, S. H. Dar, and Z. M. Elbarbary, "Empowering Data-Driven Load Forecasting by Leveraging Long Short-Term Memory Recurrent Neural Networks," *Heliyon*, vol. 10, no. 24, e40934, 2024.

[9] S. Mirasçı, S. Uygur, and A. Aksoy, "Advancing Energy Efficiency: Machine Learning-Based Forecasting Models for Integrated Power Systems in a Food Processing Company," *International Journal of Electrical Power & Energy Systems*, vol. 165, p. 110445, Apr. 2025.

[10] R. B. Bandaru, "Stacked Hybrid Model for Load Forecasting: Integrating Transformers, ANN, and Fuzzy Logic," *Scientific Reports*, vol. 15, p. 19688, 2025.