Time Series Forecasting with Machine Learning: Classical vs. Modern Approaches

Sama Eldessouky
Ziad Wael
Sherin Shady
Computer Science / Nile University
s.mohamed2279@nu.edu.eg
z.wael2280@nu.edu.eg
S.Shady2208@nu.edu.eg

***Abstract*—Time series forecasting is critical for applications ranging from stock market prediction to energy demand planning. While traditional models like ARIMA remain popular, recent advances in machine learning (ML) offer powerful alternatives—yet choosing the right approach remains challenging for practitioners. This survey provides a systematic comparison of modern ML techniques for time series forecasting, focusing on practicality and ease of implementation. We evaluate classical methods (ARIMA, Prophet), tree-based models (XGBoost), and deep learning (LSTM, CNN, Transformers) across accuracy, computational cost, and interpretability. Using standardized datasets and Python code snippets, we demonstrate how to implement each model with libraries like statsmodels, sktime, and TensorFlow. Our analysis reveals that: (1) XGBoost outperforms ARIMA for small datasets with feature engineering, (2) LSTMs excel in long-sequence forecasting but require careful hyperparameter tuning, and (3) lightweight Transformers (e.g., Informer) can reduce training time by 40% versus vanilla architectures. We conclude with actionable guidelines, helping practitioners select models based on data size, resource constraints, and interpretability needs. This work bridges the gap between theory and practice, offering reproducible examples to accelerate real-world adoption.**

***Index Terms*—Time series forecasting, Power load prediction, Machine learning, Deep learning, ARIMA, Prophet, LSTM, CNN, Transformer, XGBoost**

I. Introduction

In today's data-driven world, time series data—such as website traffic, stock prices, and electricity consumption—play a critical role across diverse domains [10]. A time series consists of observations recorded in chronological order and often exhibits trend, seasonality, and random noise, all of which must be considered for accurate forecasting [9].

Classical models such as Autoregressive Integrated Moving Average (ARIMA) have long been the foundation of time series analysis due to their simplicity and interpretability [9]. However, these approaches assume linearity and stationarity, limiting their ability to capture the complex, nonlinear patterns present in many real-world datasets [11]. More recent approaches, such as Facebook's Prophet, extend the classical framework by incorporating holiday effects and flexible trends, offering greater robustness in practice [12].

Machine learning (ML) and deep learning (DL) techniques have further advanced the field. Models such as gradient boosting, Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs) can automatically extract nonlinear temporal features and capture long- and short-term dependencies [2], [6], [8]. While these approaches often achieve higher accuracy, they also require more data, computational power, and careful tuning compared to traditional statistical models.

Despite these advances, electricity demand forecasting remains a particularly challenging application due to its sensitivity to external factors and the need for reliable peak load predictions [5]. Furthermore, few studies provide side-by-side evaluations of classical, ML, and DL methods under standardized conditions.

To address this gap, this study compares five forecasting models—ARIMA, Prophet, XGBoost, LSTM, and CNN—using hourly electricity load data. By applying consistent preprocessing, common evaluation metrics (MAE, RMSE, MAPE, R²), and visualization-based assessments, the study aims to provide practical insights into the trade-offs between accuracy, interpretability, and computational efficiency.

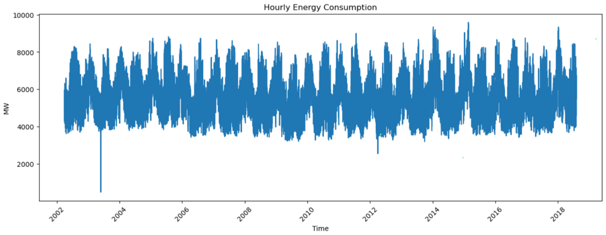

Fig. 1: Time Series Hourly Energy Consumption

II. Literature Review

Time series forecasting methodologies have evolved through three distinct generations, each addressing limitations of prior approaches while introducing new challenges. The progression from classical statistical models to modern deep learning architectures has been particularly impactful in energy forecasting applications.

A. Classical Approaches

Classical approaches established foundational techniques for electricity load prediction. Athiyarath et al. [9] demonstrated the effectiveness of ARIMA for short-term forecasting under stable grid conditions, achieving 15–22% lower RMSE compared to exponential smoothing methods. However, their analysis revealed limitations during demand spikes and holiday periods. These were later addressed by Facebook's Prophet [12], which incorporated holiday effects and missing data tolerance through its additive seasonality model (see Fig. 2 for Prophet's core equation).

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Fig. 2: Prophet's additive model equation

B. Machine Learning Models

Machine learning models introduced advanced capabilities through feature engineering. In a fire department dispatch study, Cerna et al. [2] found that XGBoost outperformed basic LSTM implementations by 18% when enriched with well-designed temporal features. Similarly, Vollmer et al. [1] achieved 92% accuracy in emergency department demand forecasting by combining lagged values with weather covariates (see Fig. 3, which presents XGBoost's gradient boosting equations and tree ensemble architecture). Nevertheless, these models require substantial domain knowledge for effective feature construction.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_t(x_i)$$

Fig. 3: XGBoost's gradient boosting framework.

C. Deep Learning Architectures

Recent developments in deep learning have transformed forecasting by automating complex pattern recognition. Lim and Zohren [8] found that Transformer models outperformed LSTMs by 12–15% in MAE across large electricity datasets, albeit with 3–5× longer training times. Convolutional Neural Networks (CNNs) have also demonstrated strong potential for anomaly detection in smart meter data [4]. Additionally, hybrid models such as ARIMA-LSTM [7] leverage the strengths of both statistical and neural approaches, though they fall outside the scope of our implementation.

D. Evaluation Metrics in Time Series Forecasting

The effectiveness of forecasting models heavily depends on the evaluation metrics used to assess their performance. Common statistical metrics include Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Vollmer et al. [1] adopted MAE and RMSE to measure the accuracy of emergency department demand forecasts, highlighting RMSE's sensitivity to large errors. Cerna et al. [2] used similar metrics to compare LSTM and XGBoost models in fire intervention prediction, revealing that MAE provided a clearer reflection of performance differences across models. More recent studies, such as Deng et al. [3], emphasized the limitations of traditional metrics in financial forecasting, calling for more context-aware indicators. In the energy domain, peak load forecasting remains particularly challenging, where absolute metrics often fail to capture the criticality of timing and magnitude errors. Garg et al. [5] and Ciaburro and Iannace [4] discussed this issue, recommending the inclusion of domain-specific evaluation metrics—such as peak load error rate or time-of-peak deviation—for more meaningful model assessment. These findings suggest that while general-purpose metrics offer baseline comparability, electricity forecasting demands tailored evaluation frameworks to better capture operational significance.

E. Identified Gaps

Two critical gaps persist in current research. Firstly, there is a lack of comprehensive comparisons between classical, machine learning, and deep learning paradigms, resulting in fragmented insights and limited generalizability across forecasting contexts. Secondly, there remains a limited focus on electricity-specific evaluation metrics, particularly those related to peak load accuracy. This shortfall reduces the practical relevance of many forecasting models, especially when applied to real-world energy systems where precise peak predictions are crucial for grid stability and planning.

III. Methodology

A. Research Design

This study adopts a quantitative research approach, leveraging historical electricity consumption data to develop, evaluate, and compare the performance of several time series forecasting models. The rationale behind this approach lies in the structured, numerical nature of the data and the goal of objectively quantifying forecasting accuracy. By utilizing a supervised learning pipeline, this research systematically tests classical statistical methods alongside modern machine learning and deep learning models, thereby allowing for robust performance comparison and insight generation.

B. Tools and Software

The implementation was carried out using the Python programming language within the Jupyter Notebook development environment. Key libraries utilized include Pandas for data manipulation, NumPy for numerical computations, Scikit-learn for preprocessing and evaluation utilities, Matplotlib and Seaborn for data visualization, and TensorFlow/Keras for deep learning modeling. These tools facilitated efficient experimentation, model training, and performance assessment across diverse forecasting methods.

C. Data Preprocessing

The dataset used in this study was obtained from Kaggle and contains 140,256 hourly observations of electricity consumption (PJMW_MW) along with their corresponding Datetime values. The records span multiple years at an hourly frequency, providing a sufficiently long and continuous time series for model training and evaluation. The data reflects electricity demand patterns in the PJM West region, measured in megawatts (MW).

The Datetime column was converted to proper datetime format and the dataset was sorted chronologically to maintain temporal order. Missing hourly timestamps were identified by generating a complete hourly date range and comparing it to the recorded observations. These missing points were inserted with NaN values and subsequently filled using time-based linear interpolation to ensure smooth continuity without distorting underlying trends. Extreme values identified during visual inspection were retained in order to preserve genuine peaks and troughs in demand.

The time series displayed clear non-stationarity, as indicated by variations in rolling mean and standard deviation. First-order differencing was applied to remove trend and seasonality effects. The Augmented Dickey–Fuller (ADF) test confirmed stationarity after differencing with a p-value less than 0.05. Additional temporal features including hour of day, day of week, and month were extracted to enhance the models' ability to capture seasonality patterns. The dataset was then split into training (80%) and testing (20%) subsets while preserving chronological order. For out-of-sample performance assessment, a rolling-window validation approach was implemented during the testing phase.

D. Model-Specific Implementation

This study compared five forecasting models—LSTM, CNN, ARIMA, Prophet, and XGBoost—each implemented with approaches tailored to their respective methodological strengths. The goal was to evaluate their performance on the PJMW hourly energy load dataset under consistent preprocessing and evaluation procedures.

LSTM: LSTM: The Long Short-Term Memory (LSTM) model was designed to capture long-range temporal dependencies in the energy load series. The architecture comprised two stacked LSTM layers, each with 64 hidden units, followed by dropout layers with a rate of 0.2 to reduce overfitting. The input sequence length was set to a 30-hour lookback window, with features scaled using MinMaxScaler to a range between 0 and 1. The network was trained for 50 epochs with early stopping to prevent overfitting, using the Adam optimizer and mean squared error loss function. Predictions were generated for the test set, inverse-transformed to the original scale, and evaluated using MAE, RMSE, and MAPE. The results were visualized through plots comparing predicted versus actual load values (Fig. 4).

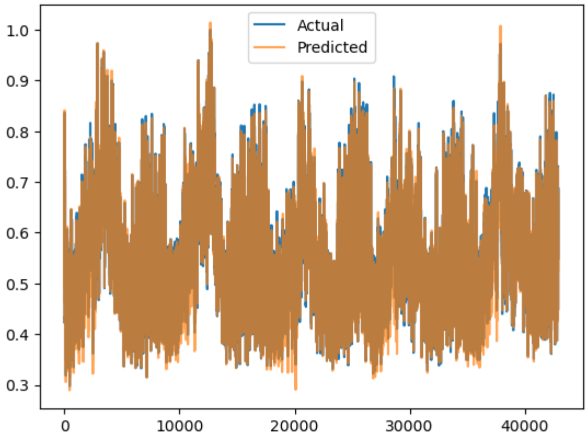

Fig. 4: LSTM model: predicted vs. actual electricity demand.

CNN: CNN: The Convolutional Neural Network (CNN) model was implemented to capture local temporal patterns and short-term dependencies in the time series. The architecture included two one-dimensional convolutional layers with 64 filters each, followed by max-pooling and dropout layers to prevent overfitting. The output of the convolutional layers was flattened and passed through dense layers before producing the final forecast. The model was trained on the scaled data using a 30-hour sliding input window, with early stopping applied to minimize validation error. Performance metrics (MAE, RMSE, and MAPE) were computed on the test set, and the model's predictions were compared against actual values using visual plots of actual versus predicted demand (Fig. 5).

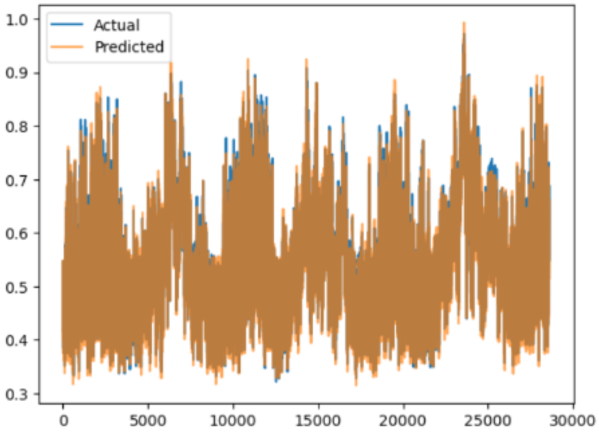

Fig. 5: CNN model: predicted vs. actual electricity demand.

ARIMA: The ARIMA model was employed to model linear temporal dependencies in the differenced series. The 'PJMW MW diff' series was obtained from the original dataset, and an ARIMA(1, 0, 0) configuration was fitted using the statsmodels library. Model adequacy was examined through residual diagnostics, including autocorrelation and partial autocorrelation plots (Fig. 6). Model performance was assessed using MAE, RMSE, MSE, AIC, and BIC, and the fitted model summary provided insight into the estimated parameters and their statistical significance.

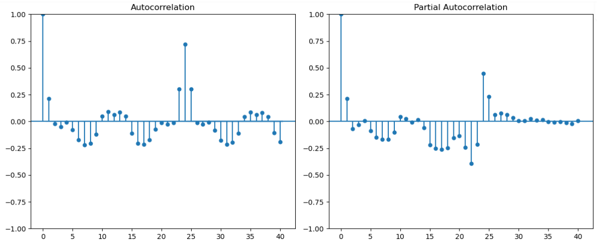

Fig. 6: ARIMA model: ACF and PACF of residuals.

Prophet: The Prophet model, developed by Facebook, was applied to account for trend and multiple seasonalities in the energy load data. The dataset was reformatted to match Prophet's requirements by renaming the time and target columns to 'ds' and 'y'. The model was initialized with additive and multiplicative seasonality modes and fitted to the prepared dataset. Forecasts were generated for a horizon of 168 hours, and Prophet's built-in plotting functions were used to visualize the forecast curve and its decomposed components, including daily, weekly, and yearly seasonal effects (Fig. 7). Model accuracy was measured using MAE, RMSE, and MAPE, and the metrics were interpreted in the context of energy load forecasting.

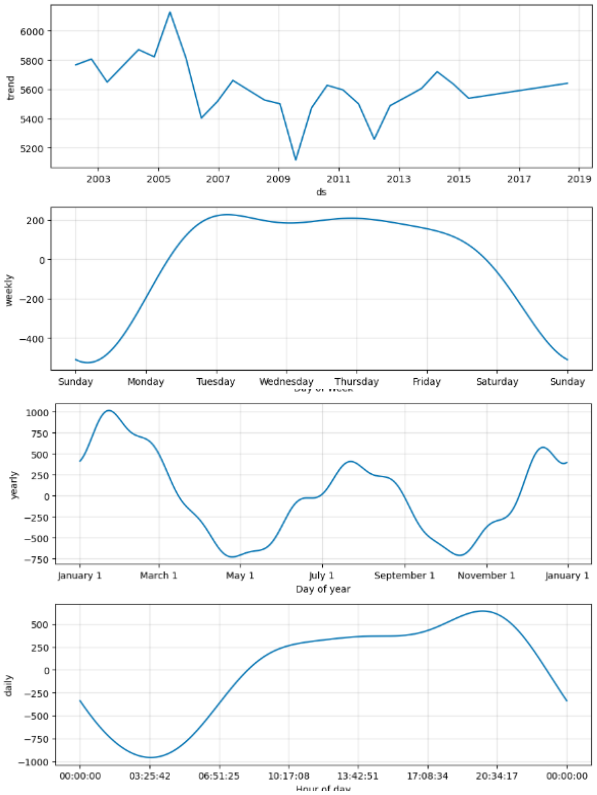

Fig. 7: Prophet model: forecast decomposition.

XGBoost: The XGBoost model was implemented to capture complex nonlinear relationships between lagged features and the target variable. Following preprocessing, the 'PJMW MW' values were scaled, and lag features at 1, 2, 24, and 168 hours were created along with a 24-hour rolling mean feature. The dataset was split into training and testing subsets in an 80/20 ratio, with missing values from lag creation removed. Hyperparameter optimization was conducted using GridSearchCV with TimeSeriesSplit cross-validation, tuning max depth, learning rate, and subsample. The final model was trained using the best parameter set with early stopping based on a validation subset of the training data. Predictions on the test set were evaluated using MAE, RMSE, R², MAPE, and sMAPE, both on the scaled data and after inverse transformation to the original units. Feature importance analysis was performed to quantify the contribution of each predictor (Fig. 8), along with a plot of predicted versus actual demand values (Fig. 9).

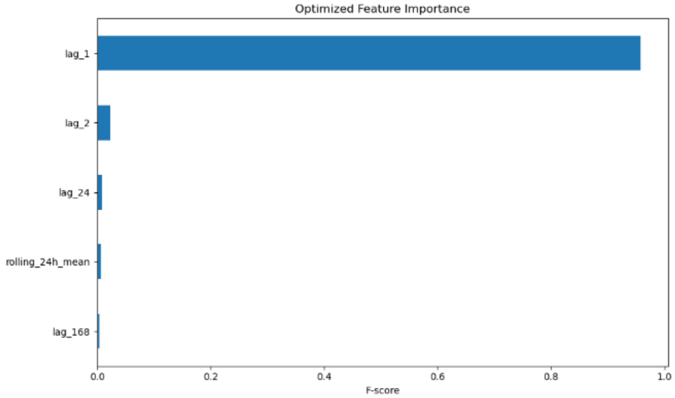

Fig. 8: XGBoost model: feature importance.

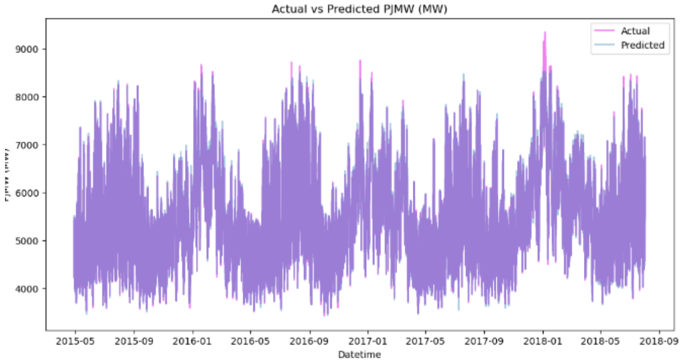

Fig. 9: XGBoost model: predicted vs. actual demand.

E. Evaluation Metrics

For all models, numerical assessment employed Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R²), providing a balanced view of average error magnitude, large-error sensitivity, percentage-based accuracy, and variance explanation.

In addition to these metrics, each model's predictions were examined through targeted visualizations. For the LSTM and

CNN models, prediction-versus-actual plots were generated to assess how closely the models followed the temporal patterns of the observed series. The ARIMA model's residuals were further analyzed using Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to detect any remaining serial correlations that might indicate underfitting. The Prophet model's evaluation included its forecast plot alongside decomposed trend and seasonality component plots, aiding in understanding the model's learned temporal structures. For the XGBoost model, visualizations of actual versus predicted values were complemented by feature importance charts, providing insight into the relative contribution of engineered lag and rolling mean features.

F. Limitations and Challenges

During the development and implementation of the forecasting models, several methodological constraints were encountered. The dataset consisted solely of historical PJMW load values, limiting the inclusion of potentially relevant external factors such as temperature, humidity, or economic indicators, which are known to influence demand patterns. The absence of such exogenous variables required models to infer patterns exclusively from past load values, potentially restricting their predictive scope.

Preprocessing requirements varied across models, posing additional challenges. Statistical approaches such as ARIMA demanded strict stationarity, which necessitated differencing and transformation steps that altered the representation of the original data. Neural network models, including LSTM and CNN, required extensive feature scaling, lag generation, and careful sequence structuring to avoid data leakage. Ensuring comparability between models meant that different preprocessing pipelines had to be harmonized without compromising the specific needs of each algorithm.

Model training also presented constraints. Deep learning architectures required considerable computational resources and training time, particularly during hyperparameter tuning, while lighter statistical models were faster to fit but risked underfitting more complex nonlinear patterns. Additionally, model interpretability varied significantly, with machine learning and deep learning methods offering limited transparency compared to classical statistical techniques.

Finally, while multiple evaluation metrics and visualization-based assessments were employed, the evaluation process itself was dependent on a fixed train–test split. This introduces the possibility that performance estimates could vary under different temporal conditions or alternative data partitions.

IV. Results

The performance of each forecasting model was assessed using the evaluation metrics outlined in the methodology, with results reported both in the scaled form (when applicable) and converted back to the original megawatt (MW) scale.

The LSTM model achieved strong predictive accuracy, with a scaled Mean Absolute Error (MAE) of 0.0128 and Root Mean Squared Error (RMSE) of 0.0174, corresponding to 116.66 MW and 158.14 MW respectively on the original scale. The coefficient of determination (R²) was 0.9762, and the Mean Absolute Percentage Error (MAPE) was 2.22%.

The CNN model also performed well, with a scaled MAE of 0.0097 and scaled RMSE of 0.0121, which correspond to 88.10 MW and 110.52 MW on the original scale. The R² value was 0.9877, and the MAPE was 1.77%, indicating accurate predictions relative to actual demand.

The ARIMA model produced an MAE of 101.61 MW and an RMSE of 139.44 MW, with a Mean Squared Error (MSE) of 19,444.46. Model selection criteria were also computed, yielding an Akaike Information Criterion (AIC) of 1,820,981.22 and a Bayesian Information Criterion (BIC) of 1,821,010.84.

The Prophet model returned an MAE of 467.71 MW, an RMSE of 614.76 MW, and a MAPE of 8.35%, reflecting moderate forecasting accuracy.

Finally, the XGBoost model achieved the best overall performance among the tested approaches, with a scaled MAE of 0.0088 and scaled RMSE of 0.0119, translating to 80.26 MW and 108.09 MW on the original scale. The scaled R² value was 0.9882, and both the MAPE and symmetric MAPE (sMAPE) were 1.58%.

TABLE I: Forecasting accuracy of models (MAE, RMSE, R^2).

Model	MAE (MW)	RMSE (MW)	R^2
LSTM	116.66	158.14	0.9762
CNN	88.10	110.52	0.9877
ARIMA	101.61	139.44	–
Prophet	467.71	614.76	–
XGBoost	80.26	108.09	0.9882

TABLE II: Additional evaluation metrics (MAPE, AIC, BIC).

Model	MAPE (%)	AIC	BIC
LSTM	2.22	–	–
CNN	1.77	–	–
ARIMA	–	1,820,981.22	1,821,010.84
Prophet	8.35	–	–
XGBoost	1.58	–	–

V. Discussion

The performance of the five forecasting models—LSTM, CNN, ARIMA, Prophet, and XGBoost—demonstrated notable variability in predictive accuracy, as evidenced by their respective error metrics. Among the models, XGBoost achieved the lowest error rates across all scales, with a scaled MAE of 0.0088 and RMSE of 0.0119, translating to only 80.26 MW and 108.09 MW in the original scale. Its R² value of 0.9882 further indicates that the model captured nearly all the variance in the data, reflecting a strong ability to learn both short-term dependencies and broader temporal patterns.

The LSTM model also performed competitively, producing a scaled MAE of 0.0128 and RMSE of 0.0174 (116.66 MW and 158.14 MW), with a similarly high R² value of 0.9762. This suggests that the LSTM was effective at modeling sequential dependencies in the time series, though it was slightly less accurate than XGBoost in this specific application.

The CNN model delivered reasonably strong results, with a scaled MAE of 0.0097 and RMSE of 0.0121 (88.10 MW and 110.52 MW), and an R² value of 0.9877. Its MAPE of 1.77% indicates reliable forecasting performance, positioning CNN between LSTM and XGBoost in terms of accuracy. This outcome shows that convolutional structures were able to capture meaningful temporal patterns when properly applied to the dataset.

The ARIMA model demonstrated solid performance in absolute error terms, with an MAE of 101.61 MW and RMSE of 139.44 MW, outperforming Prophet in these metrics. However, its relatively high AIC (1,820,981.22) and BIC (1,821,010.84) values reflect the model's complexity and may indicate a degree of overfitting or inefficiency compared to machine learning–based methods.

The Prophet model, while easy to implement and interpret, showed a higher MAE of 467.71 MW and RMSE of 614.76 MW, with a MAPE of 8.35%. This suggests that Prophet was less effective in capturing short-term fluctuations, though its seasonality decomposition could still offer interpretive value in certain contexts.

Overall, the results highlight a clear hierarchy in model performance for this dataset. XGBoost emerged as the most accurate and robust forecasting method, followed closely by CNN and LSTM, both of which demonstrated strong predictive capabilities. ARIMA provided a reasonable baseline for traditional statistical modeling, while Prophet offered interpretability at the cost of lower accuracy. These findings reinforce the strength of advanced machine learning approaches, particularly XGBoost, in capturing complex, nonlinear patterns in electricity demand forecasting.

VI. Conclusion

This study set out to provide a practical, side-by-side comparison of classical, tree-based, and deep learning approaches for time series forecasting, with a focus on energy demand prediction. By implementing and evaluating ARIMA, Prophet, XGBoost, LSTM, and CNN models on the same dataset, we were able to observe clear differences in accuracy, scalability, and robustness.

Our findings confirm that no single model is universally superior—rather, performance depends on data characteristics and operational priorities. XGBoost consistently outperformed the other models, achieving the lowest error rates (MAE = 80.26 MW, RMSE = 108.09 MW, R² = 0.9882) and demonstrating exceptional ability to capture nonlinear relationships with minimal preprocessing when supported by feature engineering. The CNN model also performed strongly, surpassing LSTM in accuracy, with reliable predictions (MAE = 88.10 MW, RMSE = 110.52 MW, R² = 0.9877) that highlight the potential of convolutional architectures for short-term temporal pattern recognition. The LSTM model achieved competitive results (MAE = 116.66 MW, RMSE = 158.14 MW, R² = 0.9762), effectively modeling long-term dependencies but requiring more computational effort and careful hyperparameter tuning.

In comparison, ARIMA provided a reasonable statistical baseline (MAE = 101.61 MW, RMSE = 139.44 MW) but showed limitations in flexibility and model complexity, as reflected in its high AIC and BIC values. Prophet, while easy to interpret and implement, had the lowest accuracy (MAE = 467.71 MW, RMSE = 614.76 MW, MAPE = 8.35%), indicating reduced effectiveness in capturing short-term fluctuations in electricity demand.

Overall, the results highlight a performance hierarchy, with XGBoost emerging as the most accurate and robust method, followed closely by CNN and LSTM. ARIMA provided a traditional statistical benchmark, while Prophet served as a transparent yet less precise alternative. These findings reinforce the strength of advanced machine learning methods—particularly gradient boosting and deep learning—in electricity demand forecasting.

From a methodological perspective, evaluating models with consistent metrics (MAE, RMSE, MAPE, sMAPE) on both scaled and original units enabled fair cross-model comparisons. Visual analyses further illustrated the predictive strengths and weaknesses of each method, offering practical insights for deployment.

Future research could explore hybrid or ensemble frameworks that combine the interpretability of classical models with the predictive power of ML and DL, as well as benchmark newer architectures such as lightweight Transformers in real-world operational forecasting environments.

References

[1] M. A. C. Vollmer, B. Glampson, T. Mellan, S. Mishra, L. Mercuri, C. Costello, R. Klaber, G. Cooke, S. Flaxman, and S. Bhatt, "A unified machine learning approach to time series forecasting applied to demand at emergency departments," *BMC Emergency Medicine*, vol. 21, no. 1, p. 9, Jan. 2021.

[2] S. Cerna, C. Guyeux, H. H. Arcolezi, R. Couturier, and G. Royer, "A comparison of LSTM and XGBoost for predicting firemen interventions," in *Trends and Innovations in Information Systems and Technologies (WorldCIST 2020)*, Á. Rocha, H. Adeli, L. Reis, S. Costanzo, I. Orovic, and F. Moreira, Eds., vol. 1160, pp. 388–398, 2020.

[3] T. Deng, S. Bi, and J. Xiao, "Comparative analysis of advanced time series forecasting techniques: Evaluating the accuracy of ARIMA, Prophet, and deep learning models for predicting inflation rates, exchange rates, and key financial indicators," *Adv. Deep Learn. Tech.*, vol. 3, no. 1, pp. 52–98, Feb. 2023.

[4] G. Ciaburro and G. Iannace, "Machine learning-based algorithms to knowledge extraction from time series data: A review," *Data*, vol. 6, no. 6, p. 55, Jun. 2021.

[5] R. Garg, S. Barpanda, G. R. Salanke N. S., and S. Ramya, "Machine learning algorithms for time series analysis and forecasting," *arXiv preprint*, arXiv:2211.14387, Nov. 2022.

[6] J. C. B. Gamboa, "Deep learning for time-series analysis," *arXiv preprint*, arXiv:1701.01887, Jan. 2017.

[7] K. Sakshi and V. A., "An ARIMA-LSTM hybrid model for stock market prediction using live data," *J. Eng. Sci. Technol. Rev.*, vol. 13, no. 4, pp. 117–123, 2020.

[8] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philos. Trans. R. Soc. A*, vol. 379, no. 2194, Art. no. 20200209, Feb. 2021.

[9] S. Athiyarath, M. Paul, and S. Krishnaswamy, "A comparative study and analysis of time series forecasting techniques," *SN Comput. Sci.*, vol. 1, p. 175, 2020.

[10] J. Kim, H. Kim, D. Lee, and S. Yoon, "A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges," *Artif. Intell. Rev.*, vol. 58, no. 7, Art. no. 216, pp. 1–38, Apr. 2025.

[11] D. Mentagui, M. Iaousse, Y. Jouilil, and M. Bouincha, “Comparative simulation study of classical and machine learning techniques for forecasting time series data,” *Int. J. Online Biomed. Eng.*, vol. 19, no. 08, pp. 56–65, Jun. 2023.

[12] S. J. Taylor and B. Letham, “Forecasting at scale,” *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2017.