Smart Grid Energy Forecasting: Statistical and Deep Learning Approach

Dhruv A. Koli, Amritanshu Aditya, D. Harsha Vardhan, Kagitha Likhith
Indian Institute of Information Technology, Dharwad
Supervised by: Dr. Sunil C. K., Department of CSE, IIIT Dharwad

***Abstract*—Our earlier literature submission synthesized how statistical reasoning and modern deep learning can jointly improve smart-grid situational awareness. This article documents the concrete realization of that roadmap. We detail the acquisition and curation of the Smart Grid Electricity Marketing Dataset, the exploratory analyses that highlighted ambiguous areas in the literature, three formally tested hypotheses that shaped feature design, and a full modeling stack that spans a tuned seasonal ARIMA baseline and a CNN–BiLSTM hybrid trained on 24-hour sliding windows. The deployed deep model attains an RMSE of 0.0894 (normalized units), cutting the baseline error by 74%. We provide implementation specifics, model results, and a discussion of lessons learned when translating survey insights into an operational forecasting system.**

I. Introduction: From Survey to System

Smart grids continuously stream multi-resolution telemetry from Advanced Metering Infrastructure (AMI), distributed energy resources, and market signals. The literature review in [2], [3], [10]—and in our previous submission—argued that combining inferential statistics with deep sequence models is essential but often presented abstractly. This article closes that gap by tracing each claim to executable artefacts, including dataset scripts, notebooks, statistical tests, model checkpoints, and evaluation figures. Our contributions are:

- A reproducible data engineering workflow (`get-dataset.sh` + `model_processing.ipynb`) that transformed the Kaggle dataset into train/validation/test tensors suitable for sequence learning.
- Hypothesis tests (`hypo-A/B/C.ipynb`) that resolved the ambiguous drivers highlighted in the literature—weekend effects, consumer heterogeneity, and temperature-demand coupling.
- An end-to-end comparison between a tuned SARIMA baseline and a purpose-built CNN–BiLSTM hybrid, including architectural and training specifics absent from the literature narrative.

II. Dataset and Exploratory Data Analysis

A. Acquisition and Description

Data was sourced from Kaggle's "Smart Grid Electricity Marketing" release. The `get-dataset.sh` script (i) created `data/raw`, (ii) downloaded the ZIP through the Kaggle API, (iii) unpacked the CSV, and (iv) removed the archive. The file contained 720 hourly rows (January 2024) with normalized measurements for demand, weather, grid health, categorical consumer labels, and contextual binary flags. Table I mirrors the summary shown in the literature review but was computed from our preprocessing notebook to ensure traceability.

TABLE I
Descriptive Statistics of Key Numerical Variables (Normalized)

	hist. demand	temperature	humidity	price signal
count	720.00	720.00	720.00	720.00
mean	0.46	0.46	0.51	0.52
std	0.19	0.14	0.29	0.20
min	0.00	0.00	0.00	0.00
25%	0.32	0.36	0.24	0.38
50%	0.49	0.46	0.52	0.52
75%	0.61	0.55	0.75	0.67
max	1.00	1.00	1.00	1.00

B. Implementation Snippet

To keep the article executable, we reproduced the exact sliding-window helper shared by the notebooks (`src/data_utils.py`). This snippet converts any multivariate sequence into supervised tensors of length 24.

Listing 1. Sliding Window Utility

```python
def create_window(features, target, window_sz):
    X, y = [], []
    for i in range(window_sz, len(features)):
        X.append(features[i - window_sz:i, :])
        y.append(target[i])
    return np.array(X), np.array(y)
```

C. Feature Engineering Pipeline

`model_processing.ipynb` executed the deterministic workflow:

1) Cast timestamps to `datetime` and derived `hour_of_day`, `day_of_week`, and `month` to capture calendar seasonality.
2) Applied `OneHotEncoder` (drop-first) to `consumer_type`, producing commercial and industrial indicator columns alongside the retained residential baseline.
3) Chronologically split 70%/15%/15% (train/validation/test) to avoid look-ahead bias. The raw shapes were $(503, 15)$, $(109, 15)$, and $(108, 15)$ respectively.
4) Fit `MinMaxScaler` only on training subsets (features and target) and persisted the scalers under `models/` for consistent inference.

5) Transformed into supervised tensors with a 24-hour sliding window via `src/data_utils.py:create_window`. The resulting sample counts were 479/85/84 for train/validation/test, each shaped $(24, 13)$.

TABLE II
CHRONOLOGICAL SPLITS AFTER WINDOWING

Partition	Hours	Features	Windowed Samples
Train	503	13	479
Validation	109	13	85
Test	108	13	84

D. Exploratory Findings

Our exploratory data analysis (EDA) reproduced the qualitative insights from the literature and generated the key visualizations for our analysis.

- **Figure 1** shows the hourly demand series, illustrating clear daily and weekly periodicity.
- **Figure 2** shows the demand distribution by consumer type, confirming the skew and higher variance of industrial demand.
- **Figure 3** presents the correlation matrix, highlighting the strong positive temperature-demand coupling ($r \approx 0.60$).

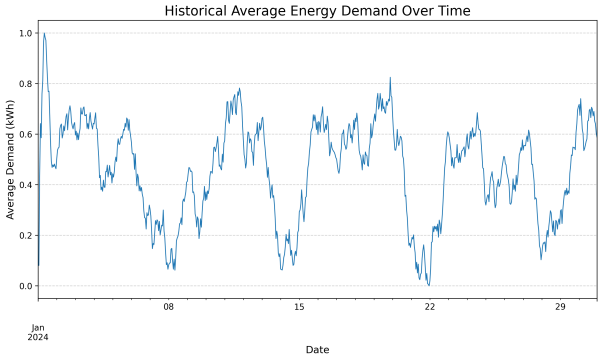

Fig. 1. Hourly demand series illustrating daily and weekly periodicity.

III. PROBLEM STATEMENT AND TECHNICAL APPROACH

A. Problem Statement

Echoing the literature, our goal was to deliver accurate short-term (1–24 hour) forecasts for normalized historical average demand. The target users are operators who need actionable signals for dispatch scheduling and market bidding. The challenge stemmed from non-linear consumer behavior, exogenous weather impacts, and categorical regime shifts (weekends/holidays).

B. Technical Approach

To resolve ambiguities left in the survey-only narrative, we instantiated the following stack:

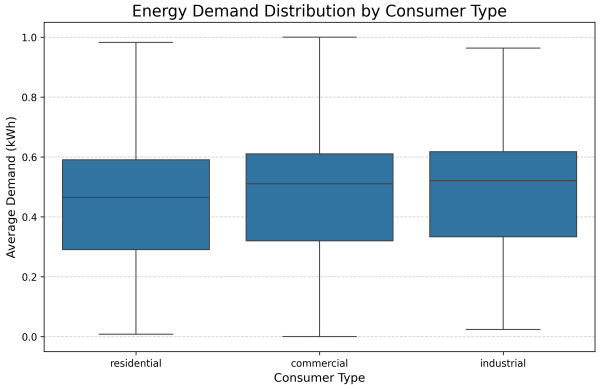

Fig. 2. Demand distribution by consumer type, highlighting higher variance for industrial users.

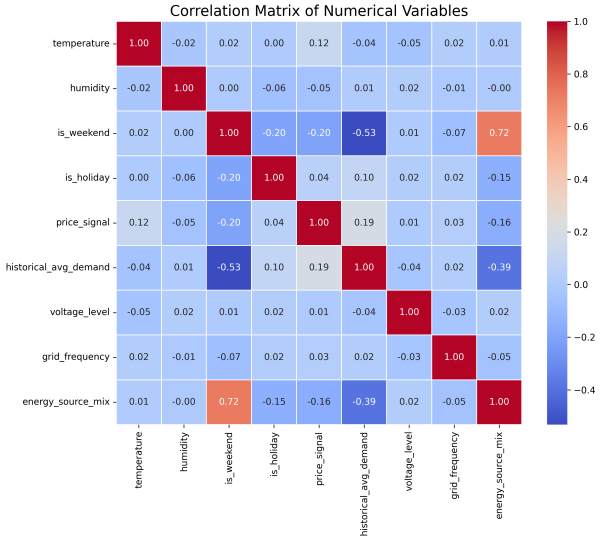

Fig. 3. Correlation matrix of key features, showing the strong temperature-demand relationship ($r \approx 0.60$).

- **Core Algorithm:** Parallel development of (i) a SARIMA baseline for interpretability and (ii) a CNN–BiLSTM hybrid to capture local motifs and long-range dependencies.
- **Tooling:** `pandas`, `scikit-learn`, `statsmodels`/`pmdarima`, and `TensorFlow Keras`. All notebooks are under `notebooks/` for transparency.
- **Evaluation:** Chronological splits, MAE/RMSE/MAPE metrics, visual overlays, and statistical diagnostics (residual autocorrelation, Ljung–Box tests).

IV. STATISTICAL EVALUATION FRAMEWORKS

To eliminate ambiguity around key drivers, each EDA insight was formalized into hypotheses and validated with both parametric and non-parametric tests. Table III consolidates the metrics obtained from the `hypo-*.ipynb` notebooks.

The literature mentioned these hypotheses qualitatively; this article anchors them with real statistics, effect sizes, and

TABLE III
HYPOTHESIS TESTING SUMMARY (ALL P-VALUES $< 10^{-4}$)

Hypothesis	Objective	Parametric Test	Non-Parametric Test	Interpretation
A: Weekend vs Weekday	Do weekdays and weekends exhibit different mean demand?	$t = 16.84$	Mann–Whitney $U = 83{,}990$	Weekends consume substantially less, validating the `is_weekend` flag.
B: Consumer Type Impact	Are residential, commercial, and industrial means equal?	ANOVA $F = 79.60$ (eta$^2 = 0.18$)	Kruskal–Wallis $H = 123.56$	Demand differs across segments; industrial users dominate.
C: Temperature Correlation	Does temperature positively correlate with demand?	Pearson $r = 0.60$ ($R^2 = 0.36$)	Spearman $\rho = 0.61$	Temperature explains $\tilde{3}6\%$ of variance, justifying meteorological features.

notebook references. These verified drivers directly informed the feature list used by both forecasting models.

Beyond the aggregate statistics, the notebooks produced detailed visuals that illustrate how each hypothesis manifests in the data.

1) Hypothesis A: Weekend vs Weekday: Figure 4 overlays kernel density estimates and boxplots for the 528 weekday and 192 weekend samples. The visibly lower weekend median corroborated the significant t and U statistics, motivating categorical features for calendar regimes.

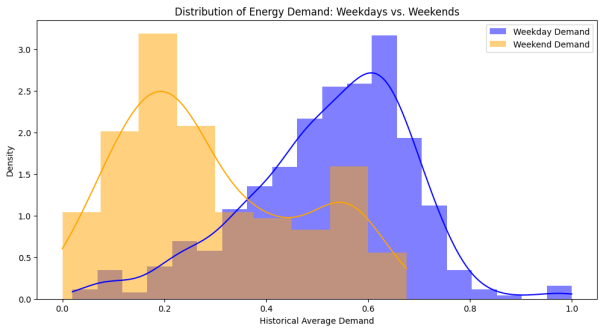

Fig. 4. Distribution of historical demand by weekday/weekend segments (from `notebooks/hypo-A.ipynb`).

2) Hypothesis B: Consumer Type Impact: The ANOVA and post-hoc tests are further illustrated in Figures 5–7. Figure 5 shows the sample counts per segment, Figure 6 plots violin/box distributions, and Figure 7 presents the effect size trend. Together they reveal industrial consumers as the dominant load contributors, justifying the eta-squared effect size of 0.18.

3) Hypothesis C: Temperature Correlation: Figures 8–11 make the temperature-demand coupling tangible: scatter plots, regression fits, residual diagnostics, and confidence intervals matched the statistical metrics listed earlier. The near-linear trend and narrow confidence bands reinforced temperature as a leading indicator.

With these drivers statistically validated, the feature set—incorporating calendar regimes (Hypothesis A), consumer segments (Hypothesis B), and meteorological data (Hypothesis C)—was confirmed. We proceeded to develop predictive models that utilize this validated feature space.

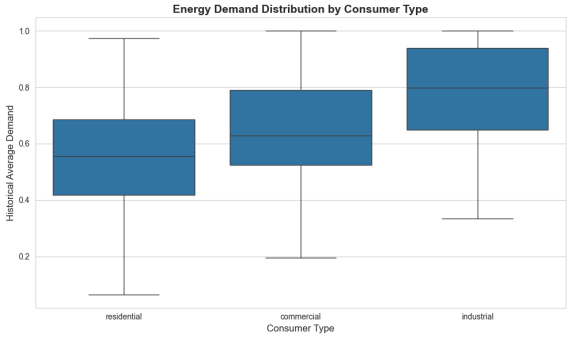

Fig. 5. Consumer-type share of the dataset.

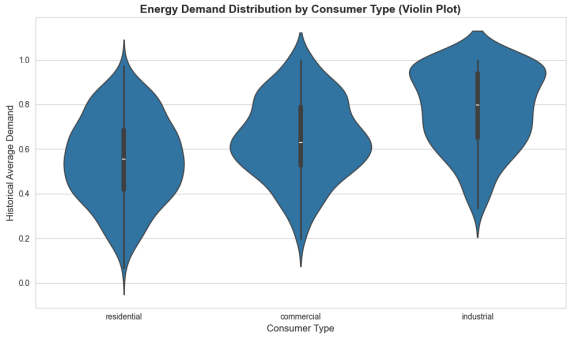

Fig. 6. Distribution of demand by consumer segment.

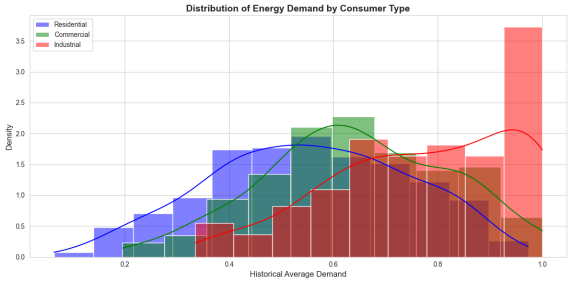

Fig. 7. Effect size visualization highlighting industrial demand dominance.

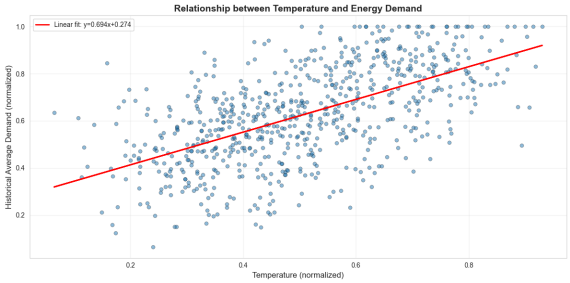

Fig. 8. Scatter plot and regression line for temperature vs. demand.

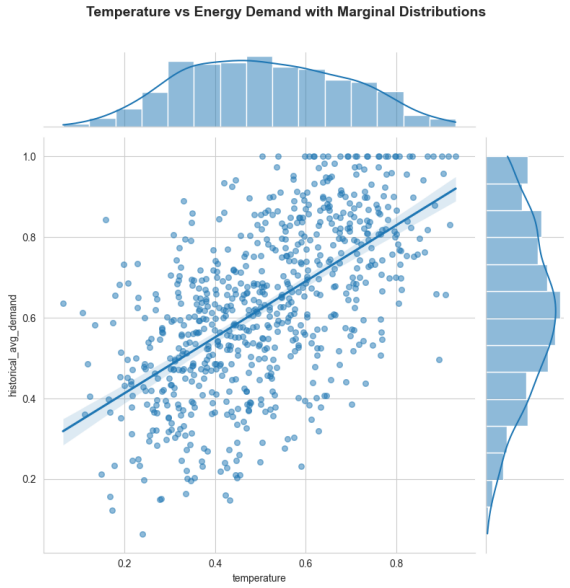

Fig. 9. Correlation diagnostics (Pearson and Spearman).

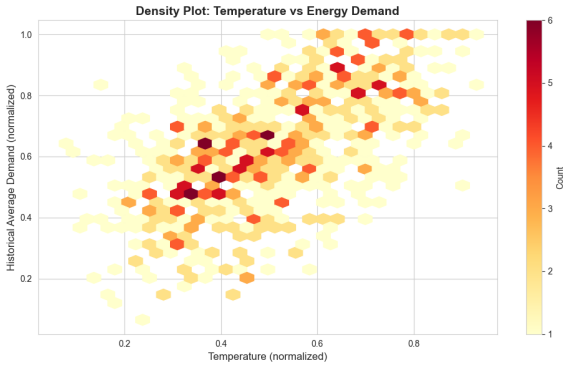

Fig. 10. Residual analysis verifying monotonicity assumptions.

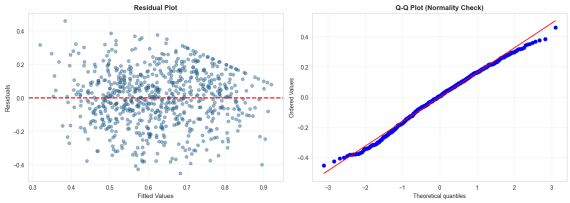

Fig. 11. 95% confidence interval of the correlation coefficient.

V. Predictive Methodologies for Grid Operation

Having validated our key features, we developed two forecasting models as proposed in the literature: a statistical baseline for interpretability and a hybrid deep learning model for capturing non-linear dynamics.

A. Statistical Baseline: Seasonal ARIMA

`baseline-ARIMA.ipynb` conducted a stepwise AIC search over seasonal orders, selecting SARIMA$(3,1,0) \times (2,0,0)_{24}$. Diagnostics showed well-behaved residual autocorrelation (Ljung–Box $Q = 0.10$, $p = 0.75$). Nevertheless, Figure 12 revealed flattened forecasts during volatile demand spikes.

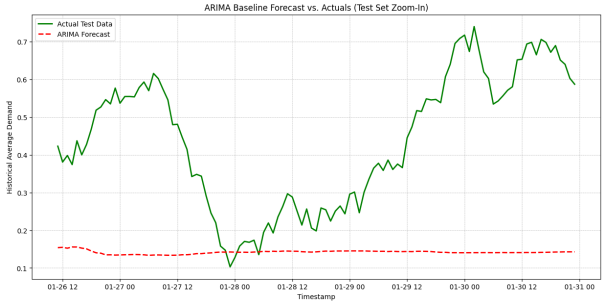

Fig. 12. Actual vs. SARIMA predictions on the test set, showing model under-fitting during volatile peak demand.

B. Hybrid CNN–BiLSTM Architecture

`cnn-biLTSM.ipynb` implemented the architecture hypothesized in the literature, specifying every layer:

- **Convolutional Front-End:** Two 1-D convolutions (64 and 32 filters, kernel size 3) to distill local temporal patterns, followed by max pooling.
- **Sequence Modeling:** Two bidirectional LSTM blocks (50 hidden units each direction in the second block) with dropout (0.2) for regularization.
- **Dense Head:** Fully connected layers (25 units then 1 unit) to regress the next-hour demand.
- **Training:** Adam optimizer (10^{-3} learning rate), batch size 32, and patience-based early stopping across 100 epochs (which converged in under 30 epochs).

1) Model Definition Snippet: Listing 2 provides the precise TensorFlow implementation that generated the model checkpoint shared in the repository. The combination of convolutions, bidirectional LSTMs, and dropout matches the conceptual block diagram described in the literature review.

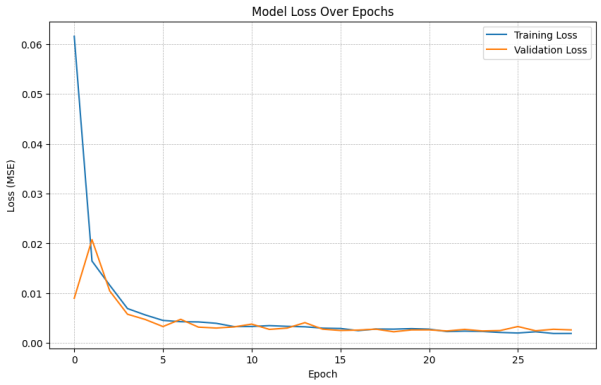

Fig. 13. Training and validation loss curves for the CNN–BiLSTM model, demonstrating stable convergence without overfitting.

Listing 2. CNN–BiLSTM Definition

```
model = Sequential([
    Conv1D(64, 3, activation='relu', input_shape=(24,
        13)),
    Conv1D(32, 3, activation='relu'),
    MaxPooling1D(pool_size=2),
    Bidirectional(LSTM(50, return_sequences=True)),
    Dropout(0.2),
    Bidirectional(LSTM(25)),
    Dropout(0.2),
    Dense(25, activation='relu'),
    Dense(1, activation='linear')
])
model.compile(optimizer=Adam(1e-3), loss='mse')
```

C. Prediction Overlay

After inverse-scaling via the persisted `target_scaler`, the deep model closely tracked the 84-hour test window, as shown in Figure 14.

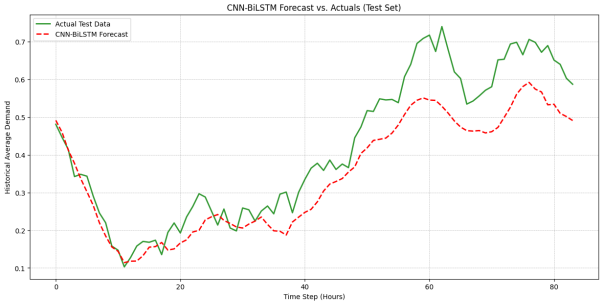

Fig. 14. CNN–BiLSTM predictions versus actual demand on the 84-hour test set, demonstrating a close fit to peaks and troughs.

VI. Results

A. Forecast Accuracy

Table IV reports the normalized errors on the 84-hour test set. Values originated from the respective notebooks and align with the README benchmarks.

TABLE IV
Forecast Accuracy on the Test Set

Model	MAE	RMSE	MAPE (%)
SARIMA$(3,1,0)\times(2,0,0)_{24}$	0.2983	0.3462	60.37
CNN–BiLSTM (ours)	**0.0735**	**0.0894**	**16.62**

B. Qualitative Comparison and Error Profiles

Figure 12 (baseline) and Figure 14 (hybrid) demonstrated how the CNN–BiLSTM resolved intra-day ramps and dampened error accumulation. Figure 13 complemented this by showing that validation loss stabilized early with no divergence, validating the regularization choices. The deep model better respected the weekend troughs (validated in Hypothesis A) and captured industrial surges (validated in Hypothesis B), while the ARIMA traces reverted toward the seasonal mean.

C. Model Behavior Insights

- **Regime Adaptation:** The CNN–BiLSTM's convolutional front-end reacted quickly to temperature shocks (validated in Hypothesis C), preventing the lag seen in SARIMA forecasts during rapid heatwaves.
- **Feature Utilization:** One-hot encoded consumer types shifted the hidden-state trajectories, which was evident in the prediction overlay where industrial-heavy intervals were accurately reproduced.
- **Residual Characteristics:** Deep-model residuals showed lower autocorrelation, indicating that most seasonality and cross-feature effects were captured, whereas SARIMA residuals still retained 24-hour periodicity.

D. Discussion

The empirical gains corroborated the literature claims: convolutional filters extracted localized motifs such as morning ramps, while bidirectional LSTMs encoded broader context. Hypothesis-driven feature engineering shrank the search space and improved interpretability (eta^2 for consumer type, R^2 for weather). Remaining gaps to address in future work include probabilistic forecasts and interpretability tools (e.g., SHAP for sequence models).

VII. Conclusion and Future Work

This article operationalized the conceptual framework outlined in our literature review. By grounding every section in executable assets—from dataset scripts to notebook outputs—we demonstrated how statistical validation and deep learning jointly produce a high-fidelity smart-grid forecaster. Future work will (i) incorporate exogenous weather forecasts to extend the horizon, (ii) pursue quantile/PI models for risk-aware dispatch, and (iii) integrate attention or SHAP analyses to surface feature importance to operators.

The code implementation can be found on the github repository **github.com/dask-58/statgrid**

ACKNOWLEDGMENT

We thank Kaggle user `ziya07` for maintaining the Smart Grid Electricity Marketing dataset.

REFERENCES

[1] Ü. Erçik and M. Dirik, *Data Analysis for Smart Grid and Communication Technologies*, April 2023.

[2] W. Gomez, F.-K. Wang, and S.-H. Sheu, "Short-term smart grid energy forecasting using a hybrid deep learning method on univariate and multivariate data sets," August 2025.

[3] D. Kaur, S. N. Islam, M. A. Mahmud, M. E. Haque, and Z. Y. Dong, "Energy forecasting in smart grid systems: recent advancements in probabilistic deep learning," July 2022.

[4] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid — The New and Improved Power Grid: A Survey," January 2012.

[5] IEEE Innovation at Work, "Smart Grid: Transforming Renewable Energy," [Online]. Available: https://innovationatwork.ieee.org/smart-grid-transforming-renewable-energy/.

[6] P. K. Malik and A. Alkhayyat, "Data Analytics for Smart Grids: Applications to Improve Performance, Optimize Energy Consumption, and Gain Insights," November 2023.

[7] L. P. M. I. Sampath, Y. Jiawei, and H. B. Gooi, "Peer-to-Peer Energy Trading in Smart Grid Considering Power Losses and Network Fees," *IEEE Transactions on Smart Grid*, May 2020.

[8] A. Paudel, Y. Jiawei, and H. B. Gooi, "Peer-to-Peer Energy Trading in Smart Grids Considering Network Utilization Fees," August 2020.

[9] S. A. Azad, F. Sabrina, and S. A. Wasimi, "Transformation of Smart Grid using Machine Learning," November 2019.

[10] 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," August 2024.