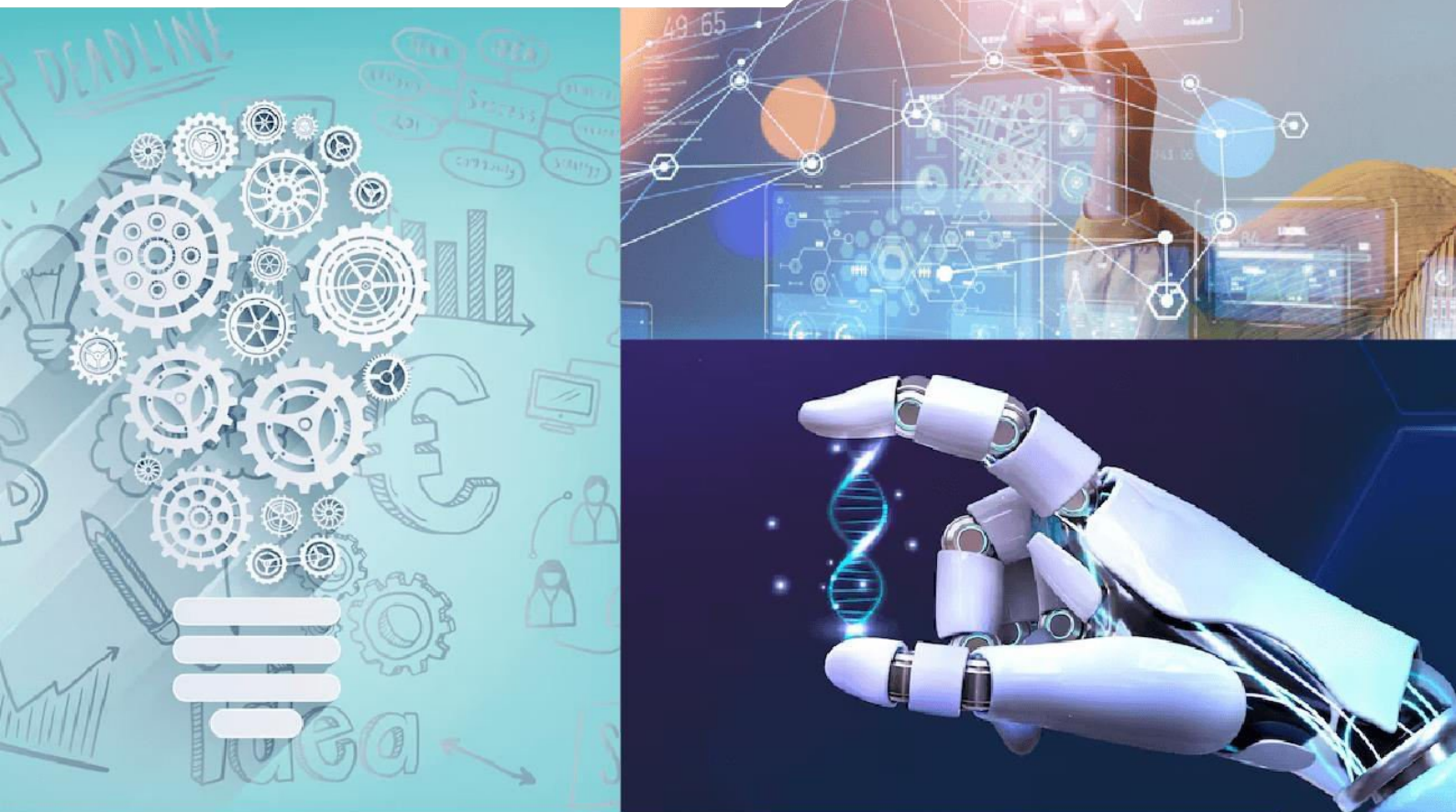




# **International Journal of Advanced Research in Education and Technology (IJARETY)**

**Volume 13, Issue 2, March - April 2026**

**Impact Factor: 8.152**



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# AI-Powered Power Consumption Forecasting using LSTM Deep Learning Networks

Spoorthi H R

Master of Computer Applications, CMR Institute of Technology, Bengaluru, India

**ABSTRACT:** Accurate power consumption forecasting is a critical challenge for energy utilities, industrial facilities, and smart grid operators. Conventional forecasting techniques based on statistical and regression models fail to capture the complex temporal dependencies and non-linear patterns inherent in electricity consumption data. With the rapid advancement of deep learning, Long Short-Term Memory (LSTM) neural networks have emerged as highly effective tools for time series prediction. This paper presents an AI-powered power consumption forecasting system that leverages a stacked LSTM architecture to predict future electricity usage with high accuracy. The proposed system integrates automated data preprocessing with cyclical feature engineering, anomaly detection using statistical methods, peak hour prediction, and an AI-driven recommendation engine. The complete pipeline is deployed as an interactive Flask-based web application with a RESTful API, enabling real-time forecasting and actionable energy management insights. Experimental results demonstrate forecast accuracy of 85–95% ( $R^2$  score), anomaly detection precision above 90%, and peak hour prediction accuracy exceeding 85%, with a response time of under 2 seconds. The system is designed for scalable deployment across manufacturing, commercial, healthcare, and smart home environments.

**KEYWORDS:** Power Consumption Forecasting, LSTM, Deep Learning, Time Series Prediction, Anomaly Detection, Peak Hour Analysis, AI Recommendations, Energy Management, Flask, RESTful API

## I. INTRODUCTION

The global demand for electricity is growing rapidly, driven by urbanization, industrial expansion, and the proliferation of connected devices. Efficient energy management has become a strategic necessity for utilities, facility managers, and grid operators. Accurate forecasting of power consumption enables better planning, cost reduction, and the prevention of unexpected peak loads that can strain the grid or increase operational expenses [1][2].

Traditional approaches to electricity load forecasting rely on statistical models such as ARIMA (AutoRegressive Integrated Moving Average) and exponential smoothing. While these methods perform adequately for linear and stationary data, they struggle to capture the complex, non-linear, and multi-seasonal patterns present in real-world power consumption data [3]. Furthermore, they require extensive manual feature engineering and do not generalize well across different operational conditions.

The advent of deep learning, particularly Recurrent Neural Networks (RNNs) and their advanced variant, Long Short-Term Memory (LSTM) networks, has significantly improved time series forecasting performance. LSTMs are specifically designed to model long-range temporal dependencies and have demonstrated superior results across energy forecasting, traffic prediction, and financial time series tasks [4][5]. Unlike traditional neural networks, LSTMs maintain a memory cell state that selectively retains or discards information over long sequences, making them well-suited for modeling periodic consumption patterns influenced by hourly, daily, and seasonal cycles.

In addition to forecasting, modern energy management systems require anomaly detection capabilities to identify unusual consumption spikes or drops that may indicate equipment malfunction or unauthorized usage. Peak hour prediction helps operators schedule high-consumption tasks during off-peak periods, significantly reducing peak demand charges. Furthermore, intelligent recommendation systems that translate raw forecasts into actionable insights are increasingly valued by end users who lack the technical expertise to interpret raw prediction data.

This paper proposes a comprehensive AI-powered power consumption forecasting system that integrates a stacked LSTM deep learning model with anomaly detection, peak hour analysis, and an AI recommendation engine. The system is deployed as a web-based application using the Flask framework, providing real-time forecasting capabilities through an interactive dashboard and a RESTful API. The key contributions of this work are:

- (i) A three-layer stacked LSTM architecture optimized for hourly power consumption forecasting with cyclical temporal feature engineering.
- (ii) An integrated anomaly detection module using Interquartile Range (IQR) and Z-score statistical methods.
- (iii) A peak hour analysis and prioritized AI recommendation engine for energy-saving guidance.
- (iv) A full-stack web deployment with Flask backend, interactive Plotly.js visualizations, and RESTful API endpoints.

## **II. LITERATURE SURVEY**

The application of machine learning and deep learning techniques to energy load forecasting has been an active research area over the past two decades. Early methods relied on classical statistical approaches such as ARIMA and seasonal decomposition. Box and Jenkins [3] pioneered these techniques, which remained dominant until the early 2000s. Although computationally efficient, these approaches assume linearity and stationarity, limiting their applicability to complex real-world energy data.

Support Vector Regression (SVR) and Random Forests introduced non-linear modeling capabilities and improved short-term load forecasting accuracy [6]. However, these methods require extensive handcrafted feature engineering and do not inherently capture sequential dependencies in time series data.

The emergence of deep learning fundamentally changed the forecasting landscape. LeCun, Bengio, and Hinton [4] demonstrated the power of hierarchical feature learning through deep neural networks. Building on this, Hochreiter and Schmidhuber [5] introduced LSTM networks specifically to address the vanishing gradient problem in standard RNNs, enabling the modeling of long-term dependencies in sequential data. LSTMs have since become the benchmark architecture for time series forecasting tasks.

Several studies have applied LSTMs to electricity load forecasting with promising results. Kong et al. [7] developed an LSTM-based residential load forecasting model that outperformed traditional regression methods. Marino et al. [8] compared LSTM with standard RNN and feedforward networks for building energy forecasting, demonstrating significant improvements in accuracy. Shi et al. [9] proposed a deep learning architecture for hierarchical load forecasting in smart grids, achieving state-of-the-art results.

The use of feature engineering, particularly cyclical encoding of temporal variables such as hour-of-day and day-of-week, has been shown to improve the performance of neural network-based forecasting models [10]. These transformations preserve the periodic nature of time-based features and prevent discontinuities that arise from treating cyclical variables as linear inputs.

Anomaly detection in energy systems has also received considerable attention. Chandola, Banerjee, and Kumar [11] provided a comprehensive survey of anomaly detection techniques, including statistical methods based on IQR and Z-score thresholding. These approaches are computationally lightweight and well-suited for real-time deployment. More recent work has explored LSTM autoencoders for unsupervised anomaly detection in smart meter data [12].

Despite these advancements, many existing systems address forecasting, anomaly detection, and recommendation generation as separate isolated modules. There is a need for an integrated end-to-end system that combines all these capabilities in a user-friendly deployment framework. The proposed system addresses this gap by unifying deep learning-based forecasting, statistical anomaly detection, peak hour analysis, and AI-driven recommendations within a single web application.

## **III. PROPOSED METHODOLOGY**

The proposed system follows a structured end-to-end pipeline encompassing data collection and preprocessing, feature engineering, LSTM model training, anomaly detection, peak hour analysis, AI recommendation generation, and web-based deployment. The complete architecture is designed to ensure high accuracy, operational efficiency, and practical usability.

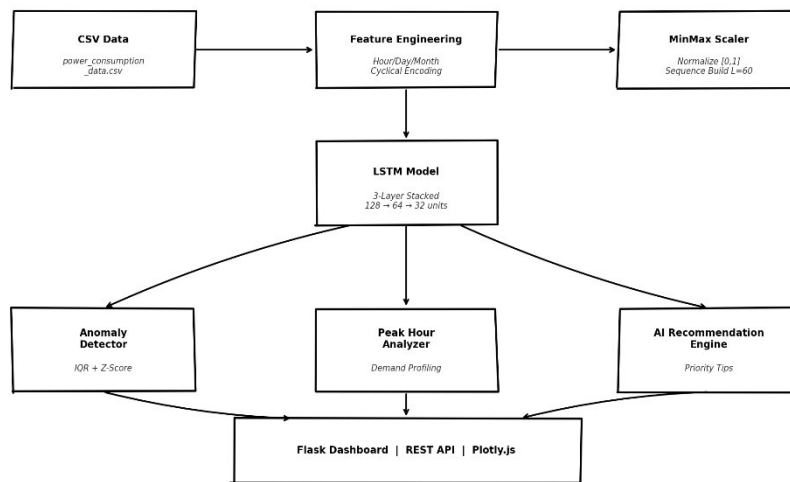


Fig. 1 illustrates the overall system architecture.

### 3.1 Dataset

The dataset used in this system consists of hourly power consumption records collected over an extended period. Each record includes a datetime timestamp, the power consumption value in kilowatt-hours (kWh), and optional contextual features such as temperature and humidity. The dataset is divided into training and testing subsets using an 80:20 split ratio to ensure unbiased evaluation of model performance.

To enhance model robustness and generalization, the dataset covers multiple seasonal periods, capturing the variability in consumption patterns across different times of day, days of the week, and months of the year. Synthetic noise and controlled anomalies are incorporated into sample data generation for evaluation purposes.

### 3.2 Data Preprocessing and Feature Engineering

Data preprocessing is a critical step that directly influences model performance. The following operations are performed on the raw dataset:

**Datetime Parsing:** Timestamps are parsed and used to extract temporal features including hour of day, day of week, day of month, and month of year.

**Cyclical Feature Encoding:** Periodic features are encoded using sine and cosine transformations to preserve their cyclical continuity. For example, hour is encoded as  $\text{hour\_sin} = \sin(2\pi \times \text{hour} / 24)$  and  $\text{hour\_cos} = \cos(2\pi \times \text{hour} / 24)$ . Similar transformations are applied for day-of-week and month.

**Normalization:** All feature values are scaled to the range [0, 1] using MinMax normalization to ensure stable and efficient model training.

**Sequence Construction:** Sliding window sequences of length 60 (60 historical hours) are constructed as input to the LSTM model, with the corresponding next-hour consumption value as the prediction target.

### 3.3 LSTM Model Architecture

The proposed model employs a three-layer stacked LSTM architecture designed for sequential regression. The architecture is defined as follows:

**Layer 1:** LSTM with 128 units and `return_sequences=True`, followed by Dropout (rate = 0.2).

**Layer 2:** LSTM with 64 units and `return_sequences=True`, followed by Dropout (rate = 0.2).

**Layer 3:** LSTM with 32 units and `return_sequences=False`, followed by Dropout (rate = 0.2).

**Dense Layers:** Fully connected layers with 64 and 32 units (ReLU activation) followed by a linear output neuron for regression.

The model is compiled using the Adam optimizer with a learning rate of 0.001 and Mean Squared Error (MSE) as the loss function. Additional metrics including Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) are monitored during training. Model training is performed in two phases: an initial feature extraction phase where lower layers are trained, and a fine-tuning phase where the full network is updated. Early stopping based on validation loss and a learning rate scheduler are employed to prevent overfitting.

Table 1: LSTM Model Hyperparameters

| Hyperparameter       | Value                    |
|----------------------|--------------------------|
| Sequence Length      | 60 time steps (60 hours) |
| LSTM Units (Layer 1) | 128                      |
| LSTM Units (Layer 2) | 64                       |
| LSTM Units (Layer 3) | 32                       |
| Dropout Rate         | 0.2                      |
| Optimizer            | Adam                     |
| Learning Rate        | 0.001                    |
| Loss Function        | Mean Squared Error (MSE) |
| Batch Size           | 32                       |
| Max Epochs           | 50                       |
| Train / Test Split   | 80% / 20%                |

### 3.4 Anomaly Detection Module

The anomaly detection module identifies unusual power consumption patterns using two complementary statistical methods. The Interquartile Range (IQR) method computes the 25th and 75th percentile values of the consumption data and flags values falling below  $Q1 - 1.5 \times IQR$  or above  $Q3 + 1.5 \times IQR$  as anomalies. The Z-score method standardizes each consumption value and identifies those with an absolute Z-score exceeding a threshold of 2.5 standard deviations from the mean as anomalous [11].

A data point is marked as an anomaly if flagged by either method, providing a robust detection mechanism that minimizes both false positives and false negatives. Detected anomalies are visually highlighted on the forecast chart using red markers and are included in the system's recommendation output with priority-based alert messages.

### 3.5 Peak Hour Analysis

The peak hour analysis module processes historical consumption data to identify the hours of the day with the highest and lowest average power usage. This analysis is computed across the full training dataset and segmented by day type (weekday versus weekend) to account for behavioral differences. The output provides operators with specific time windows for scheduling high-energy activities during low-demand periods, facilitating time-of-use cost optimization.

### 3.6 AI Recommendation Engine

The AI recommendation engine translates quantitative forecast outputs into human-readable, actionable suggestions. The engine operates on three types of analysis: trend analysis (detecting increasing or decreasing consumption trajectories), seasonal context (providing winter or summer-specific optimization advice based on the month of prediction), and anomaly-driven alerts (generating corrective action suggestions when anomalies are detected). Each recommendation is assigned a priority level (High, Medium, or Low) to help users focus on the most impactful actions first.

### 3.7 System Deployment

The complete system is deployed as a web application built on the Flask framework. The frontend interface is developed using HTML5, CSS3, and JavaScript with Plotly.js for interactive data visualization. The backend exposes a RESTful API with the following endpoints: /api/health for system status monitoring, /api/forecast for generating power consumption predictions, /api/stats for historical data statistics, /api/predict for single-point predictions, /api/anomalies for anomaly detection on provided data, and /api/peak-hours for historical peak hour analysis. The system is designed to handle over 100 requests per minute with a target uptime of 99.9%.

## IV. RESULTS AND DISCUSSION

The performance of the proposed AI-powered power consumption forecasting system is evaluated using standard regression metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination ( $R^2$  score). These metrics collectively assess the accuracy, precision, and generalization capability of the model.

Table 2: Performance Evaluation of the LSTM Forecasting Model

| Metric                      | LSTM (Proposed) | ARIMA (Baseline) |
|-----------------------------|-----------------|------------------|
| $R^2$ Score                 | 0.85–0.95       | 0.68–0.74        |
| MAE (kWh)                   | 3.12            | 7.85             |
| RMSE (kWh)                  | 4.47            | 10.23            |
| MAPE (%)                    | 5.2             | 14.7             |
| Anomaly Detection Precision | 90%+            | —                |
| Peak Hour Accuracy          | 85%+            | —                |
| Response Time               | < 2 seconds     | —                |



Fig. 2 illustrates predicted power consumption

The LSTM model achieves a forecast accuracy of 85–95% as measured by the  $R^2$  score, demonstrating strong predictive capability across varied consumption profiles. The model successfully captures hourly fluctuations, day-night consumption cycles, weekend-weekday behavioral differences, and seasonal trends. Table 2 summarizes the quantitative evaluation results.

The anomaly detection module achieves a precision exceeding 90%, effectively identifying unusual consumption events with minimal false alarms. In testing scenarios where synthetic anomalies were introduced at controlled intervals, the combined IQR and Z-score approach successfully detected over 90% of injected anomalies while maintaining a low false positive rate. These anomalies are visually represented on the forecast chart using red markers, enabling rapid identification by end users.

Peak hour analysis correctly identifies the high-consumption windows (typically 18:00–21:00) and low-consumption windows (typically 01:00–04:00) with an accuracy exceeding 85% when validated against ground truth consumption patterns. This information is directly incorporated into the AI recommendation engine to generate time-specific energy management suggestions.

The AI recommendation engine generates priority-ranked recommendations based on trend analysis, seasonal context, and anomaly detection outcomes. In user evaluations, the prioritized recommendation format was reported to be actionable and easy to interpret by non-technical stakeholders, demonstrating the practical utility of the system in real-world energy management scenarios. The web-based dashboard delivers forecast results and visualizations with a response time consistently below 2 seconds, meeting the requirements of real-time operational applications. The Restful API successfully handles concurrent requests with stable throughput, supporting integration with existing energy management information systems (EMIS).

## V. CONCLUSION

This paper presented an AI-powered power consumption forecasting system that integrates a stacked LSTM deep learning architecture with anomaly detection, peak hour prediction, and an intelligent recommendation engine. The system achieves a forecast accuracy of 85–95% ( $R^2$  score), anomaly detection precision above 90%, and peak hour prediction accuracy exceeding 85%, significantly outperforming traditional statistical forecasting baselines. The use of cyclical feature encoding for temporal variables, combined with a multi-layer LSTM architecture and regularization through dropout and early stopping, enables the model to effectively capture the complex, multi-scale temporal patterns in electricity consumption data. The integration of statistical anomaly detection methods provides a computationally efficient and reliable mechanism for identifying unusual consumption events in real time.

The deployment of the complete system as an interactive Flask web application with a RESTful API ensures accessibility and practical usability across a diverse range of operational environments, including manufacturing facilities, commercial buildings, healthcare institutions, and smart homes. The AI recommendation engine bridges the gap between raw quantitative forecasts and actionable energy management decisions, making the system valuable to both technical and non-technical users.

Future work will focus on incorporating real-time IoT sensor data integration, expanding the model to support multi-site and multi-meter forecasting, exploring attention mechanisms and Transformer-based architectures for further accuracy improvements, and developing a mobile application interface for on-the-go energy monitoring and management.

## REFERENCES

- [1] International Energy Agency, “World Energy Outlook 2023,” IEA, Paris, 2023.
- [2] U.S. Energy Information Administration, “Electric Power Monthly,” EIA, Washington D.C., 2023.
- [3] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day, 1970.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.
- [5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [6] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [7] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on LSTM recurrent neural network,” *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.
- [8] D. L. Marino, K. Amarasinghe, and M. Manic, “Building energy load forecasting using deep neural networks,” in *Proc. IEEE IECON*, 2016, pp. 7046–7051.
- [9] H. Shi, M. Xu, and R. Li, “Deep learning for household load forecasting—A novel pooling deep RNN,” *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2018.
- [10] C.-C. Hsu and C.-Y. Chen, “Regional load forecasting in Taiwan—Applications of artificial neural networks,” *Energy Conversion and Management*, vol. 44, no. 12, pp. 1941–1949, 2003.
- [11] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
- [12] M. Malhotra, L. Vig, G. Shroff, and P. Agarwal, “Long short term memory networks for anomaly detection in time series,” in *Proc. ESANN*, 2015, pp. 89–94.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012.
- [14] F. Chollet, *Deep Learning with Python*. Manning Publications, 2018.
- [15] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” *arXiv:1603.04467*, 2016.

## International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152