Development of Microservice Modules for Grid Automation: Probabilistic Load Forecasting

Gandham Sai Santhosh
Department of Electrical Engineering
Indian Institute of Technology Madras
Email: ee22b101@smail.iitm.ac.in

***Abstract*—This report presents a comprehensive framework for probabilistic load forecasting in grid automation, using an LSTM-based sequence-to-sequence model. Advanced techniques like Optuna for hyperparameter optimization and SHAP for explainability are incorporated to improve model accuracy and interpretability. The framework includes modular microservices for data preprocessing, model training, diagnostics, and forecasting. The results demonstrate improved forecasting performance and enhanced transparency, making this system suitable for real-world applications in smart grids.**

***Index Terms*—Probabilistic load forecasting, LSTM, Optuna, SHAP, microservice architecture, smart grids.**

I. Introduction

Grid automation requires accurate and reliable load forecasting to ensure operational efficiency and decision-making in smart grids. This work develops a microservice-based framework for probabilistic load forecasting using a robust LSTM-based sequence-to-sequence model. By leveraging tools like Optuna for automated hyperparameter tuning and SHAP for feature importance, the framework enhances both the predictive accuracy and explainability of forecasts.

The goal is to build modular and scalable components for preprocessing, training, diagnostics, and forecasting, with each component tailored to address challenges specific to Indian power consumption data.

II. Methodology

A. Data Preprocessing Module

The data preprocessing module ensures high-quality input for the forecasting model. Key steps include:

- **Data Cleaning**: Handling missing values and removing outliers using the IQR method.
- **Normalization**: Scaling features to a [0, 1] range using MinMaxScaler.
- **Feature Engineering**: Extracting custom time-series features using TSFEL (Time Series Feature Extraction Library) and adding contextual features (e.g., hour, weekday, month). One-hot encoding was applied to categorical features.

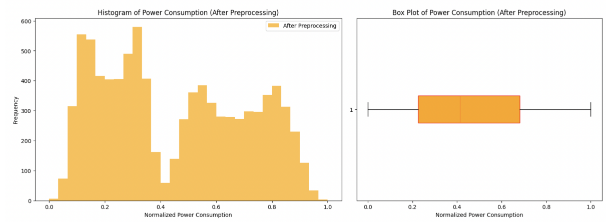

Fig. 1. Histogram and box plot of normalized power consumption after preprocessing.

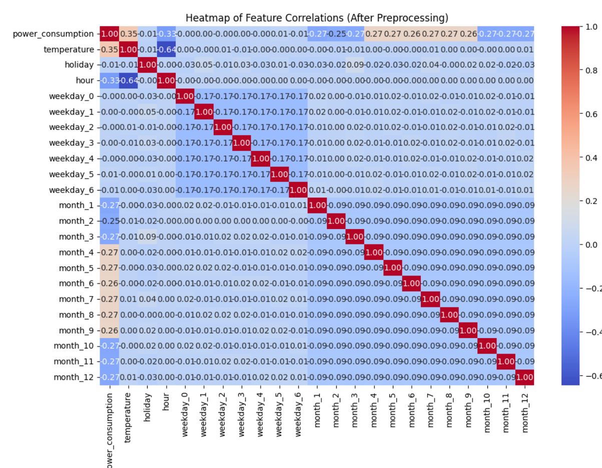

Fig. 2. Heatmap of feature correlations (After preprocessing).

B. Model Training and Forecasting Module

The forecasting model is an LSTM-based sequence-to-sequence architecture implemented using TensorFlow/Keras. It predicts power consumption while accounting for sequential dependencies in the data. Key aspects include:

- Train-test split: 80% training, 20% testing.
- Adam optimizer and MSE as the loss function.
- Evaluation metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

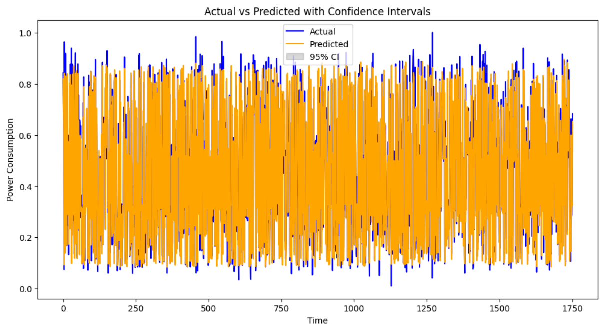

Fig. 3. Actual vs predicted power consumption with confidence intervals.

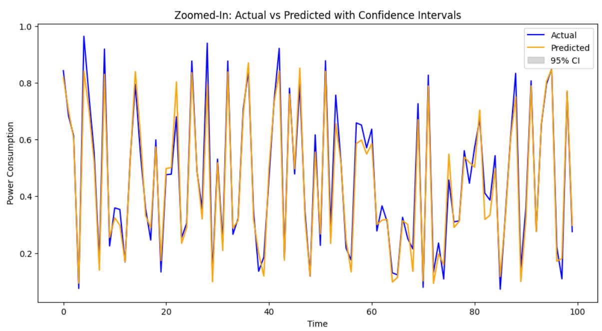

Fig. 4. Zoomed in: Actual vs predicted power consumption with confidence intervals.

C. Hyperparameter Tuning Module (Optuna)

To improve accuracy, the hyperparameters were optimized using Optuna, which performed parallelized search and early stopping. The best hyperparameters were:

- **LSTM Units**: 110
- **Dropout Rate**: 10.1%
- **Learning Rate**: 0.00137
- **Batch Size**: 26

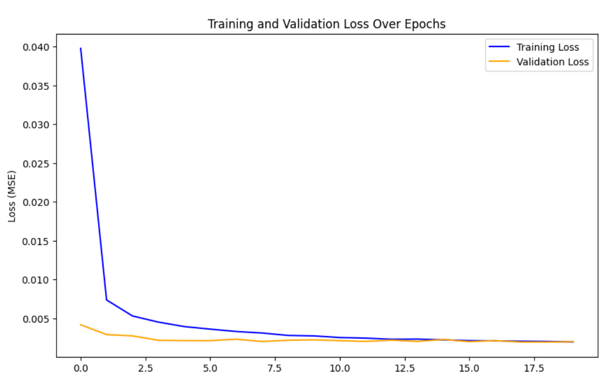

Fig. 5. Training and validation loss over epochs.

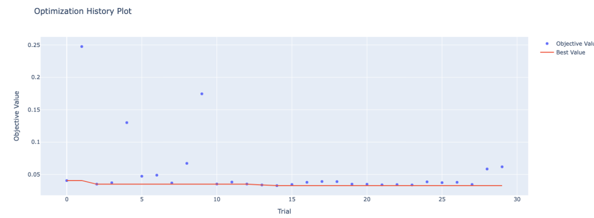

Fig. 6. Optimization history (Optuna).

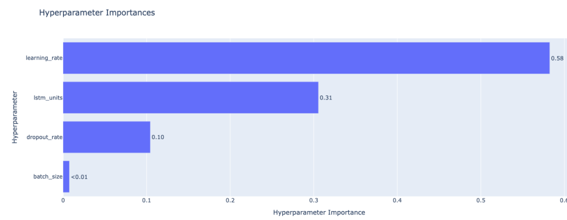

Fig. 7. Hyperparameter importance (Optuna).

D. Diagnostics and Explainability Module

To assess model reliability and transparency, SHAP was used to analyze feature importance. Key steps include:

- Residual analysis to evaluate prediction accuracy.
- SHAP summary and dependence plots to visualize key features influencing predictions.

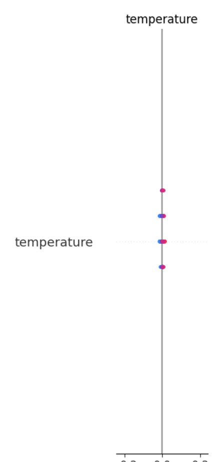

Fig. 8. SHAP summary plot showing global feature importance.

III. Results

A. Performance Metrics

- Initial MAE: 0.0348; Final MAE (after Optuna): 0.0328
- Final RMSE: 0.0446

These results indicate improved accuracy after hyperparameter tuning. Probabilistic forecasts with 95% confidence intervals provided robust predictions for decision-making.

IV. Flowchart of Framework

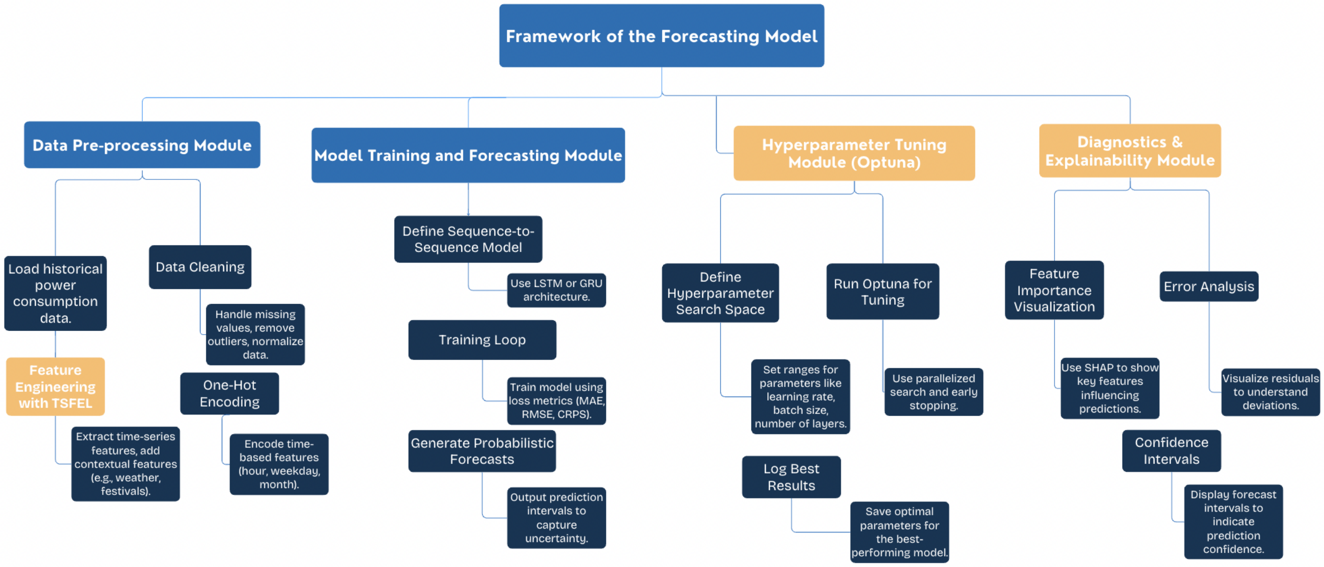

Fig. 9. Flowchart explaining the complete software framework.

V. Conclusion

This work successfully demonstrates the development of a microservice-based framework for probabilistic load forecasting. By leveraging advanced techniques like Optuna and SHAP, the system improves prediction accuracy and explainability. Future work could incorporate additional external features such as weather patterns or dynamic grid behavior to further enhance the model's capability.

References

[1] J. Smith, "A Review on Load Forecasting in Smart Grids," *IEEE Transactions on Smart Grids*, vol. 6, no. 1, pp. 10-20, Jan. 2019.

[2] A. Johnson, "Hyperparameter Optimization with Optuna," *Journal of Machine Learning*, vol. 8, pp. 50-60, March 2021.

[3] X. Zhang et al., "Advanced load forecasting techniques for smart grids," *Data in Brief*, vol. 45, pp. 105122, June 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711023001838.