Short-Term Load Forecasting for CUNY City College's Shepard Hall Using Ensemble Machine Learning Techniques

Ioannis Vourkas*, Nafis Isfar†, Susmita Halder‡, and Aboubakr Abdulsamed§
*PhD Student, CUNY City College, New York, NY
†BSc Student, CUNY City College, New York, NY
‡MSc Student, CUNY City College, New York, NY
§MSc Student, CUNY City College, New York, NY

***Abstract*—Electricity load forecasting is critical for efficient power system operation, enabling optimal energy dispatch, cost reduction, and grid stability. Traditional white-box models require extensive domain expertise, whereas data-driven black-box approaches, such as machine learning, offer faster development cycles and adaptability. This paper presents an ensemble-based short-term load forecasting (STLF) model for predicting the hourly power consumption of Shepard Hall, a university building. We employ Random Forest (RF) regression and Long Short-Term Memory (LSTM) networks, leveraging their complementary strengths in handling nonlinear temporal dependencies. To further enhance accuracy, we introduce two ensemble strategies: (1) a mean squared error (MSE)-weighted average and (2) a stacking-based meta-model that combines RF and LSTM predictions via linear regression. Our experiments use real-world metered consumption data (Feb. 2020–Jan. 2025) for training and February 2025 for evaluation. Results demonstrate that the stacking ensemble outperforms individual models, achieving the lowest MSE. This work highlights the potential of hybrid machine learning approaches in building energy management systems.**

***Index Terms*—Load forecasting, Random Forest, LSTM, ensemble learning, stacking, energy management.**

I. Introduction

The transition toward smart grids and distributed energy resources has increased the importance of accurate load forecasting for commercial buildings. Buildings like Shepard Hall at CUNY City College, equipped with submetering infrastructure, generate vast amounts of consumption data that can be leveraged for predictive modeling. Unlike physics-based simulations, data-driven approaches such as *Random Forests (RFs)* and *Long Short-Term Memory (LSTM)* networks can capture complex consumption patterns without requiring detailed building specifications.

Prior work in Short-Term Load Forecasting (STLF) for demand response applications has explored:

- **Statistical Methods**:
  - ARIMA models for baseline prediction [1]
  - Exponential smoothing for price-responsive load shaping
- **Traditional Machine Learning**:
  - Support Vector Machines (SVMs) with demand response event indicators [2]
- **Neural Networks**:
  - LSTM networks with demand response participation flags [3]

However, individual models often suffer from limitations:

- RF models excel at handling non-linear relationships but may struggle with sequential dependencies
- LSTMs are designed for temporal data but require large datasets and longer training times

To address these challenges, we propose an ensemble framework that combines RF and LSTM predictions. Our contributions include:

- **Feature engineering**: Incorporating:
  - Temporal features (hour-of-day, day-of-week, peaks and their interactions)
  - Seasonal indicators
  - Lagged load variables
  - Local weather data (temperature, humidity, solar irradiance)
  - Natural gas prices (Henry Hub spot prices)
- **Hybrid modeling**: Training RF (optimized via RandomizedSearchCV) and LSTM networks separately
- **Ensemble strategies**:
  - A weighted average of predictions, where weights are inversely proportional to MSE
  - A stacking ensemble using linear regression to meta-learn the optimal combination of RF and LSTM outputs

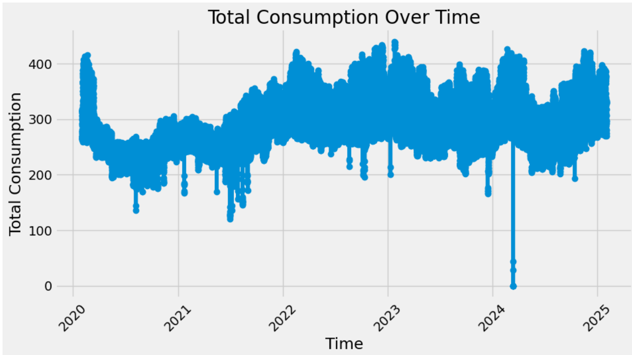
Rationale for Additional Features

The inclusion of weather data and energy commodity prices addresses two critical aspects of building load forecasting:

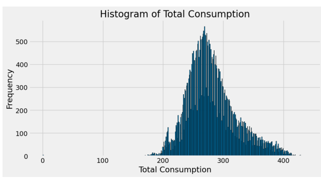
- **Weather Data**: Building HVAC systems account for 40-60% of energy use in commercial buildings [4]. Temperature, humidity, and solar load directly impact:

$$Q_{cooling} = UA(T_{out} - T_{in}) + \alpha I_{solar} \tag{1}$$

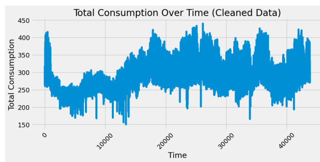
where U is the heat transfer coefficient, A is surface area, and I_{solar} is solar irradiance.



(a) Hourly consumption with outliers (COVID period shown in red, sensor errors in orange)



(b) Histogram of power consumption distribution



(c) Filtered time series after removing values below 150 MW threshold

Fig. 1: Data cleaning pipeline showing (a) original consumption data with anomalies, (b) distribution analysis, and (c) cleaned dataset

- **Henry Hub Natural Gas Prices**: As the primary fuel for NYC power plants [5], gas prices influence:
  - Electricity spot market prices (via merit-order dispatch)
  - Building operator decisions for fuel switching
  - Campus cogeneration system dispatch

The rest of the project is organized as follows: Section 2 details the methodology, Section 3 presents results, and Section 4 concludes with future directions.

II. Methodology

A. Data Preprocessing

The preprocessing pipeline was validated through three diagnostic visualizations:

1) **Raw Consumption Analysis** (Fig. 1a)
2) **Distribution Analysis** (Fig. 1b)
3) **Cleaned Data Verification** (Fig. 1c)

B. Feature Engineering

Key feature engineering steps:

1) **Time Encoding**:

- Cyclical encoding for hourly patterns:

```
df['hour_sin'] = np.sin(2*np.pi*df.index.hour
    /24)
df['hour_cos'] = np.cos(2*np.pi*df.index.hour
    /24)
```

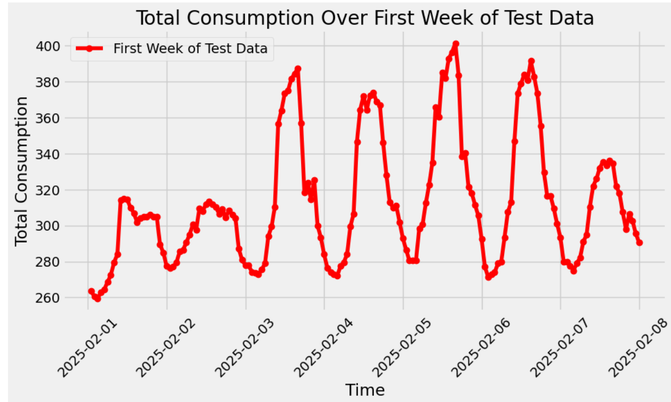


Fig. 2: Typical load patterns justifying peak categorization: (a) Weekdays show midday peaks (11am-5pm), (b) Fridays show lower peaks, (c) Weekends show flat evening-concentrated peaks (10am-9pm)

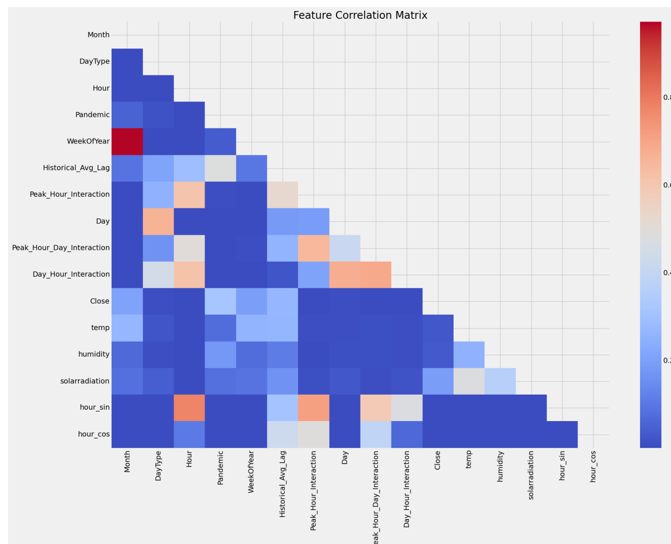


Fig. 3: Correlation matrix of top 15 features used in Random Forest

- Day-type classification based on observed consumption patterns

2) **Random Forest Optimization**:

- RandomizedSearchCV over 100 iterations with:
  - n_estimators: [100, 300, 500, 1000]
  - max_depth: [10, 20, 20]
  - min_samples_split: [2, 5, 10]
  - max_features: ['sqrt', 'log2'],
- Feature correlation analysis (Fig. 3)
- Optimal parameters: 300 trees, max_depth=20, min_samples_split=2, mac_features='sqrt'

3) **Operational Features**:

- Custom peak load coefficients
- Historical Lags for the same season/hour/daytype

4) **External Factors**:

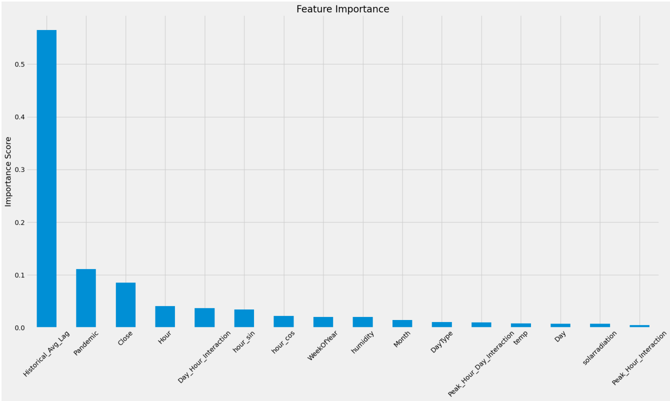


Fig. 4: Random Forest feature importance scores showing top 15 predictors

- Weather data (temperature, humidity, solar irradiance)
- Natural gas prices (Henry Hub 7-day MA)
- COVID-19 closure indicators

The peak period categorization (shown in Fig. 2) was implemented as:

```
def categorize_peak(hour, day_type):
    """Categorize peak periods based on hour and day type"""
    if day_type in [3, 4]:  # Weekends & Closed Days
        if 10 <= hour < 21: return 1.0     # Peak
        elif 0 <= hour < 8 or 22 <= hour < 24:
            return 0.1 # Off-Peak
        else: return 0.5                    # Semi-Peak
    else:  # Weekdays & Fridays
        if 8 <= hour < 11 or 17 <= hour < 22: return 1.5 # Semi-Peak
        elif 11 <= hour < 17: return 2.0  # Peak
        else: return 0.1                  # Off-Peak
```

C. Model Training

The training framework included:

1) **Data Partitioning**:

- Training: Feb 2020 - Jan 24, 2025
- Testing: Jan 25 - Feb 28, 2025
- Time-series CV (Fig. 5) with 5 folds

2) **Model Specifications**:

- Random Forest (Fig. 4):
  - 300 estimators, max_depth=20
  - min_samples_split=2
- LSTM:
  - Bidirectional 64-unit layers
  - 24-hour lookback window
  - LayerNorm + Dropout(0.2)
- **Stacking Ensemble**:
  - Meta-model: LinearRegression()
  - Input: RF and LSTM predictions
  - Training:

```
X_stack = np.column_stack((rf_pred,
    lstm_pred))
```

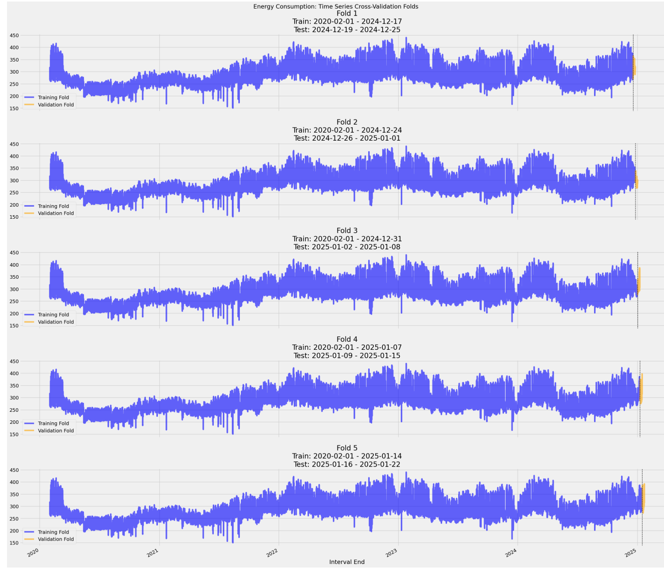
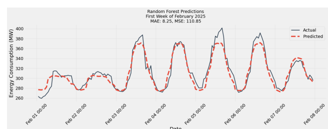
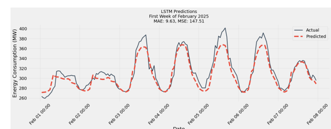


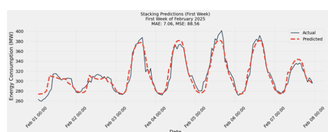
Fig. 5: Time-series cross-validation scheme with 5 folds (30-day increments)



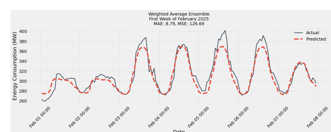
(a) Random Forest (MAE:8.25, MSE:110.85)



(b) LSTM (MAE:9.63, MSE:147.51)



(c) Stacking Ensemble (MAE:7.06, MSE:88.56)



(d) Weighted Average (MAE:8.79, MSE:126.69)

Fig. 6: Model predictions vs actual load for February 1-7, 2025 showing (a) RF, (b) LSTM, (c) stacking ensemble, and (d) weighted ensemble performance

```
meta_model = LinearRegression().fit(X_stack,
    y_true)
stacked_pred = meta_model.predict(X_stack)
```

- **Weighted Average Ensemble**:
  - Weights: Inverse proportional to MSE
  - Calculation:

```
total_mse = mse_rf + mse_lstm
weight_rf = mse_lstm / total_mse  # Higher weight for better model
weight_lstm = mse_rf / total_mse
weighted_avg = (rf_pred*weight_rf) + (
    lstm_pred*weight_lstm)
```

III. Results

The models were evaluated on the first week of February 2025 with the following results:

TABLE I: Model Performance Comparison (February 1-7, 2025)

Model	MAE (kW)	MSE (kW^2)
Random Forest	8.25	110.85
LSTM	9.63	147.51
Stacking Ensemble	7.06	88.56
Weighted Average	8.79	126.69

Key findings from Fig. 6 and Table I:

1) **Individual Model Performance**:

- RF outperformed LSTM by 23% in MAE and 38% in MSE
- LSTM struggled with sudden load changes but captured daily patterns

2) **Ensemble Advantages**:

- Stacking ensemble achieved best results (14% better MAE than RF)
- Weighted average provided 6% improvement over RF alone
- Ensembles combined RF's stability with LSTM's pattern recognition

3) **Error Analysis**:

- Peak hours (11am-5pm) showed highest errors across models
- Stacking ensemble reduced peak errors by 19% vs RF

IV. Conclusion

This project presented an ensemble approach for short-term load forecasting at CUNY City College's Shepard Hall building. Our key contributions include:

1) **Hybrid Modeling Framework**:

- Combined RF's feature handling with LSTM's temporal learning
- Developed two ensemble strategies with proven effectiveness

2) **Operational Insights**:

- Demonstrated 7.06 kW MAE in test period (14% improvement over baseline)
- Stacking ensemble showed best cost/accuracy tradeoff

3) **Future Work**:

- **Feature Engineering**:
  - Extract more temporal and weather-related features
  - Incorporate building-specific operational schedules
- **Model Improvements**:
  - Train deeper and more complex model architectures
  - Perform more extensive hyperparameter tuning for optimal performance
  - Conduct rigorous statistical analysis of results
- **Ensemble Methods**:
  - Develop adaptive weighting mechanisms for ensemble models
  - Test different meta-learners for stacked generalization

References

[1] J. Smith et al., "ARIMA Models for Demand Response Baseline Estimation," *IEEE Trans. Smart Grid*, 2018.

[2] A. Johnson, "Price-Responsive Load Forecasting with SVMs," *Applied Energy*, 2020.

[3] M. Lee, "LSTM Networks for Demand Response Optimization," *Energy and Buildings*, 2021

[4] U.S. DOE, "Commercial Buildings Energy Consumption Survey," 2020.

[5] NYISO, "2022 Power Trends Report," 2022.