**IBM SkillsBuild Hydropower Climate Optimization Challenge – First Place Solution**

### **1. Overview and Objectives**

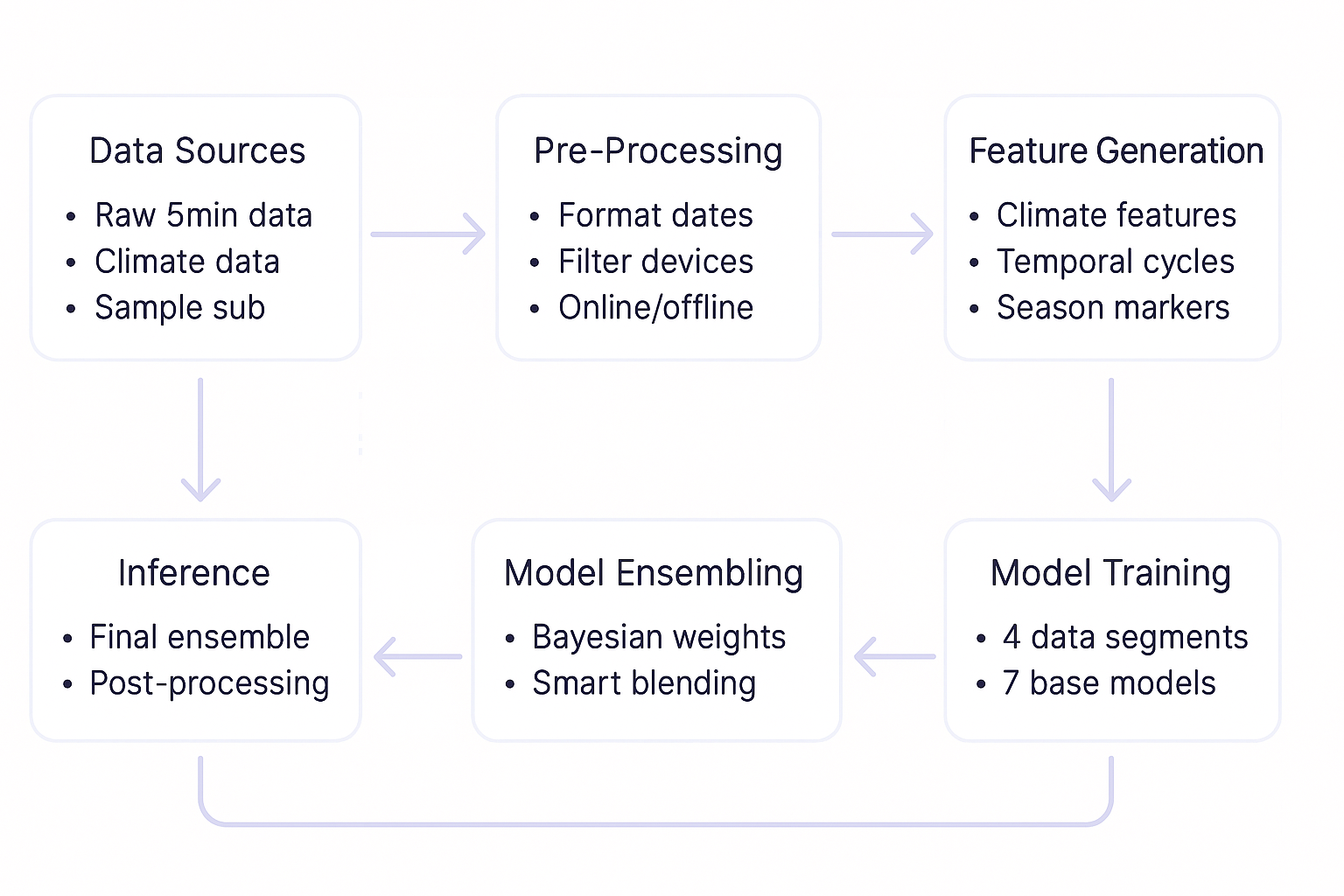
**Purpose** This document describes a winning solution for the *IBM SkillsBuild Hydropower Climate Optimization Challenge*. The goal is to predict daily energy consumption for multiple consumer devices in Kalam, Pakistan. The solution applies comprehensive feature engineering, strategic data segmentation, and advanced ensemble modeling to handle various seasonal and climate-related complexities.

**Objectives and Expected Outcomes**

1. **Accurate kWh Predictions**: Achieve low RMSE for daily consumption forecasts.
2. **Seasonality and Weather Integration**: Incorporate Pakistan’s unique seasonal and weather factors.
3. **Robustness**: Effectively manage zero-consumption periods and device filtering.
4. **Efficient Inference Pipeline**: Offer a systematic approach from raw data transformation to final predictions.

### **2. Architecture Diagram**

Below is a concise representation of how ETL, modeling, and inference steps interact:



### **3. ETL Process**

#### **3.1 Extract**

* **Data Sources:**
  + **Primary dataset** (5-min consumption logs)
  + **Climate data** (temperature, precipitation, wind components, snowfall)
  + **Sample submission file** (for test device-user alignment)
* **Data Formats:** CSV for consumption, Excel for climate
* **Extraction Considerations:**
  + Filter out devices/users not present in the test set
  + Handle potentially large volume of 5-min data but aggregated to daily

#### **3.2 Transform**

1. **Data Cleaning:**
   * Timestamps → Pandas datetimes
   * Identify “online” vs. “offline” periods based on weekly zero usage
2. **Aggregation:**
   * Convert 5-minute data → daily sum (kwh\_sum) plus mean, std, min, max for voltage/current/power factor
3. **Feature Engineering:**
   * Extract consumer/device IDs from Source
   * Add cyclical temporal features (day of year, day of week, season, sin/cos transformations)
   * Merge in climate data (temperature means, extremes, precipitation, wind speed)
   * Mark major Pakistani holidays and Ramadan intervals
   * Rolling/exponential smoothing for temperature trends (acceleration, volatility)

#### **3.3 Load**

* **Output Files:**
  + **filtred\_online\_days\_enriched.csv**: final daily dataset (no zero-consumption weeks, relevant users)
  + **offline\_days\_enriched.csv**: offline intervals/days
* These CSVs serve as inputs to the modeling step.

### **4. Data Modeling**

#### **4.1 Strategic Data Segmentation**

The dataset is split into four subsets to capture season-driven behaviors:

1. **Data1**: August–September 2024 & October 2023 (late summer/fall, **high** consumption, close to test period)
2. **Data2**: November–December 2023 & July 2024 (winter + mid-summer, **moderate** consumption)
3. **Data3**: Remaining intervals; often **zero or minimal** consumption
4. **Data4**: Full dataset for a **global** view

#### **4.2 Model Training**

**Seven LightGBM configurations** handle various hyperparameters:

1. **Precise**: Lower learning rate (0.01), deeper trees (max\_depth=8)
2. **Feature Selective**: Aggressive feature fraction (0.7), smaller colsample (0.6)
3. **Robust**: Outlier-friendly with higher reg\_alpha, moderate max\_depth (5)
4. **Deep Forest**: Very deep (max\_depth=12), many estimators (n\_estimators=3000)
5. **Highly Regularized**: reg\_alpha=2.0, reg\_lambda=2.0 to combat overfitting
6. **Fast Learner**: Higher learning rate (0.08) for rapid convergence
7. **Balanced**: A bias-variance trade-off approach (bagging\_freq=5, min\_child\_weight=12)

**Evaluation & Optimization**

* **5-Fold Cross-Validation**: Applied per segmented dataset
* **Bayesian Optimization**: Finds the best weights to combine these 7 base models

### **5. Inference**

1. **Filtering & Preprocessing**: The same logic used to create daily “online” data is run on new data points.
2. **Model Ensembling**: Weighted predictions from each base model (the 7 LightGBMs) are aggregated using Bayesian-optimized weights.
3. **Updates & Versioning**: The entire pipeline can be retrained or updated if new devices, new years, or additional climate variables appear.

### **6. Run Time**

* **Notebook Execution**: ~25–30 minutes end-to-end (CPU environment).
* **Model Training**: The majority of the time is spent training 7 LightGBM configurations with 5-fold CV plus Bayesian weight optimization.

### **7. Performance Metrics**

* **Primary Metric**: Root Mean Squared Error (RMSE)
* **Private Leaderboard RMSE**: **4.312706649**
* **Final Rank**: **1st Place**
* Additional metrics:  
  + **OOF RMSE**: helps monitor potential overfitting.
  + ~4% improvement gained from Bayesian weighting vs. naive ensemble averaging.

### **8. Error Handling and Logging**

* **Logging**: Configured at the start of the notebook with logging library (INFO level).
* **Error Handling**: If a given base model fails in training, a placeholder LightGBM model (n\_estimators=100) is substituted, preventing pipeline interruptions.

### **9. Maintenance and Monitoring**

* **Monitoring**:  
  + Re-check offline vs. online classification each season
  + Verify climate data merges for future expansions
* **Scaling**:  
  + Potential device-based sub-model expansions if data volumes grow
* **Lifecycle Management**:  
  + Retraining scheduled for new seasons or if patterns shift significantly
  + Potential integration with specialized time-series models (ARIMA, advanced gradient boosting, more feature engineering)

### **10. Conclusion and Performance Summary**

**Highlights**

1. **Strategic Data Segmentation**: Dividing data by distinct seasonal patterns yields specialized sub-models.
2. **Advanced Feature Engineering**: Climate, cyclical time, holiday, and temperature-trend features capture critical consumption dynamics.
3. **Ensemble Diversity**: Seven distinct LightGBM configs reduce overfitting, ensuring robust final predictions.
4. **Bayesian Optimization**: Delivers an additional ~4% improvement in ensemble synergy vs. equal-weight blending.

**Key Results**

* **Private Leaderboard RMSE**: 4.312706649
* **Final Rank**: 1st

**Future Improvements**

1. **Device-Specific Modeling**: Per-device or per-user sub-models.
2. **Time-Series Integration & Expanded Modeling**: ARIMA, advanced gradient boosting approaches, deeper feature engineering.
3. **Zero-Consumption Handling**: More nuanced treatment for extended zero-usage intervals.
4. **Extreme Weather Modeling**: Additional features or specialized sub-models for severe temperature events.

Overall, this approach—combining domain-informed feature engineering and a carefully optimized ensemble—demonstrates robust, high-accuracy predictions for energy consumption in a hydropower context.

**Notebook Name**: first\_place\_solution  
 **Training Time**: ~25–30 minutes (full run).