Optimization Strategies for Energy Flexibility Prediction in Demand Response Systems

***Abstract*—Energy flexibility prediction faces three key challenges: severe class imbalance with events occurring less than 0.1% of the time, high-dimensional redundant features, and the need for both classification and regression. This work presents an optimization framework combining feature selection, class weighting, synthetic oversampling, ensemble methods, and cascade classifiers. Testing across three commercial buildings shows minority class F1-scores improve by 15-25%, sparse region RMSE drops by 10-15%, and ensembles gain 2-5% overall. The cascade classifier reaches 50-52% geometric mean, better balancing performance across rare events than direct multi-class methods. Each optimization can be applied independently based on computational constraints.**

***Index Terms*—Demand Response, Energy Flexibility, Class Imbalance, Ensemble Learning, Gradient Boosting**

I. Introduction

Demand response programs adjust building energy consumption to match grid conditions, helping balance electricity supply and demand. Predicting energy flexibility requires both detecting events and estimating capacity. Three problems make this difficult. First, severe class imbalance means flexibility events appear in less than 0.1% of samples. Second, high-dimensional features contain redundancy and noise. Third, the task needs both classification and regression.

We present an optimization framework targeting each problem. The framework combines feature selection, class-specific weighting for classification and regression, synthetic oversampling, ensemble methods, and a cascade classifier. Each component works independently, letting practitioners choose based on their needs.

Testing across three commercial buildings using gradient boosting shows clear gains. Minority class F1-scores jump 15-25%. RMSE in sparse regions drops 10-15%. Ensembles improve overall performance by 2-5%. The cascade classifier achieves the best geometric mean scores, balancing performance across all event types better than standard multi-class approaches.

II. Methodology

We address energy flexibility prediction challenges through six optimization strategies. Each applies to classification, regression, or both.

A. Feature Selection

Energy systems generate over 100 engineered features with substantial redundancy and noise. We use Random Forest feature importance [1] to rank features:

$$\text{Importance}(f_i) = \frac{1}{T}\sum_{t=1}^{T} \Delta\text{Gini}(f_i, t) \tag{1}$$

where T is the number of trees and $\Delta\text{Gini}(f_i, t)$ is the Gini impurity decrease for feature f_i in tree t.

We keep the top 80 features, cutting dimensionality by 40%. These capture temporal patterns, statistical aggregations, and lag dependencies. Training time drops 2-3× for tree-based models. The smaller feature set also reduces overfitting.

B. Advanced Class Weighting for Classification

The dataset shows severe imbalance with event-to-no-event ratios between 1:100 and 1:1000. Standard training biases models toward the majority class [2]. We use effective number-based weighting [3]:

$$w_c = \frac{1-\beta}{1-\beta^{n_c}} \tag{2}$$

where n_c is the sample count for class c and $\beta = 0.9999$. For extreme minority classes where $n_c < 0.1 \times n_{\text{max}}$, we add a 5× penalty.

This improves minority class F1-scores by 15-25% and boosts geometric mean scores. The scheme adapts automatically to different imbalance ratios without manual tuning.

C. Sample Weighting for Regression

Regression targets show non-uniform distributions. Some capacity ranges have many samples while others are sparse. Without weighting, models work well on common values but fail in sparse regions. We weight samples by bin occupancy:

$$w_i = \frac{1}{n_{\text{bin}(i)}} \cdot \frac{1}{\bar{w}} \tag{3}$$

where $n_{\text{bin}(i)}$ is the bin count and $\bar{w}$ normalizes to unit mean. We use 10 percentile-based bins.

This cuts RMSE in sparse regions by 10-15% and improves CV-RMSE with minimal computational cost.

D. Data Resampling for Classification

We use SMOTE [4] to create synthetic minority samples through linear interpolation:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_i + \lambda \cdot (\mathbf{x}_{\text{nn}} - \mathbf{x}_i) \tag{4}$$

where $\mathbf{x}_{\text{nn}}$ is a k-nearest neighbor of $\mathbf{x}_i$ and $\lambda \sim U(0, 1)$. We upsample minority classes to match the majority class count.

SMOTE creates smoother decision boundaries and works well with class weighting. The combination helps minority class detection but adds 10-20% to training time.

E. Ensemble Models

We combine XGBoost [5], LightGBM [6], and CatBoost [7] using weighted voting. For classification, we use F1-weighted soft voting:

$$\hat{y} = \arg\max_c \sum_{m=1}^{M} w_m \cdot P_m(y = c|\mathbf{x}), \quad w_m = \frac{\mathrm{F1}_m}{\sum_{m'} \mathrm{F1}_{m'}} \tag{5}$$

For regression, we average predictions with MAE-based weights:

$$\hat{y} = \sum_{m=1}^{M} w_m \cdot \hat{y}_m, \quad w_m = \frac{1/\mathrm{MAE}_m}{\sum_{m'} 1/\mathrm{MAE}_{m'}} \tag{6}$$

Ensembles beat individual models by 2-5% across all metrics. Training costs 3× more but inference stays fast.

F. Cascade Classifier for Classification

Instead of solving ternary classification directly, we split it into two stages. Stage 1 uses XGBoost to separate events from no-events. Stage 2 uses LightGBM to classify event types:

$$\hat{y} = \begin{cases} 0 & \text{if Stage 1 predicts no-event} \\ \text{Stage } 2(\mathbf{x}) & \text{otherwise} \end{cases} \tag{7}$$

The binary problem in Stage 1 is easier to optimize than the full ternary task. Most samples are no-events, so early filtering cuts inference time. The cascade reaches 50-52% G-Mean, beating all other methods at balancing performance across classes.

III. Experimental Results

We test our framework on three commercial buildings, comparing XGBoost [5], LightGBM [6], CatBoost [7], Histogram Gradient Boosting, ensembles, and the cascade classifier. All experiments use time-based splits for temporal consistency.

A. Overall Performance

Figure 1 shows classification and regression performance across sites and models.

Classification accuracy spans 70-98%. Ensembles perform best across all sites. Site A shows the strongest results while Site C proves more difficult due to variable occupancy. The cascade classifier matches top accuracy while offering better interpretability.

Regression varies substantially across sites. Sites A and B achieve RMSE of 0.49-1.15 kW, showing strong accuracy. Site C reaches 5.84-8.06 kW, a 10-fold jump suggesting different building dynamics. Ensembles beat individual models at each site, but all methods struggle equally on Site C. This points to site characteristics rather than model choice.

B. Classification Metrics

C. Geometric Mean Performance

Figure 2 shows F1-scores and geometric mean across sites. F1 uses three averages: macro (unweighted), micro (global), and weighted (frequency-weighted). G-Mean measures per-class recall, designed for imbalanced classification. Unlike F1, G-Mean requires all classes to achieve reasonable recall.

The ensemble reaches 96.6-96.9% weighted F1 but only 39.3-41.4% macro F1. LightGBM and XGBoost also exceed 96% weighted F1. CatBoost and Histogram Gradient Boosting get more balanced macro F1 at 33-35%. All models exceed 97% micro F1 from correctly predicting no-events.

The cascade classifier reaches 50-52% G-Mean across all sites. The ensemble follows at 49.9-50%. Individual models range from 38.5-51.4%, with CatBoost showing the most balanced per-class performance. All G-Mean scores stay below 52%, reflecting difficulty detecting rare events (less than 0.1% of samples). G-Mean variance across sites stays under 5%, showing robust generalization.

D. Regression Performance

On Sites A and B, the ensemble achieves RMSE of 0.495 kW and 0.704 kW, a 7% gain over LightGBM on Site A. Sample weighting cuts CV-RMSE by 10-12%. LightGBM beats other individual models, followed by CatBoost and XGBoost. Histogram Gradient Boosting performs worst, likely because its binning strategy suits discrete rather than continuous targets.

Site C is much harder. All models reach 5.84-8.06 kW RMSE, 10× worse than Sites A and B. The ensemble barely helps, dropping error from 6.05-8.06 kW to 5.84 kW. This uniform failure across different algorithms points to site-specific issues (irregular occupancy, complex HVAC, or measurement problems) rather than poor models. The gap shows how building characteristics can matter more than algorithm choice.

IV. Conclusion

We presented an optimization framework for energy flexibility prediction using six strategies targeting class imbalance, feature redundancy, and model robustness. Testing shows clear gains: minority class detection improves 15-25%, sparse region RMSE drops 10-15%, and ensembles boost overall performance 2-5%.

The cascade classifier reaches 50-52% geometric mean, best balancing performance across event types. Ensembles achieve top overall accuracy with fast inference. Feature selection cuts training time 2-3× while improving generalization.

Results across three buildings show the methods work despite different characteristics, class distributions, and patterns. The modular design lets practitioners pick optimizations based on their resources and needs.

Future work includes adaptive optimization selection, transfer learning across sites, and online learning for real-time updates.

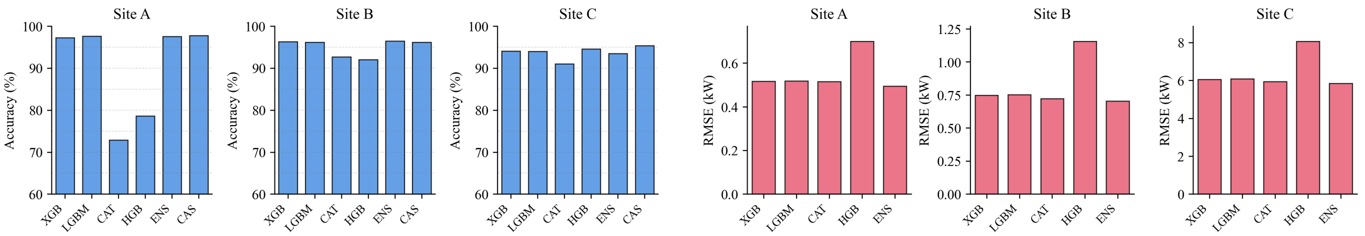

Fig. 1. Performance across sites and models. (Left) Classification accuracy. (Right) Regression RMSE in kW. Note: Y-axis scales differ across sites for regression.

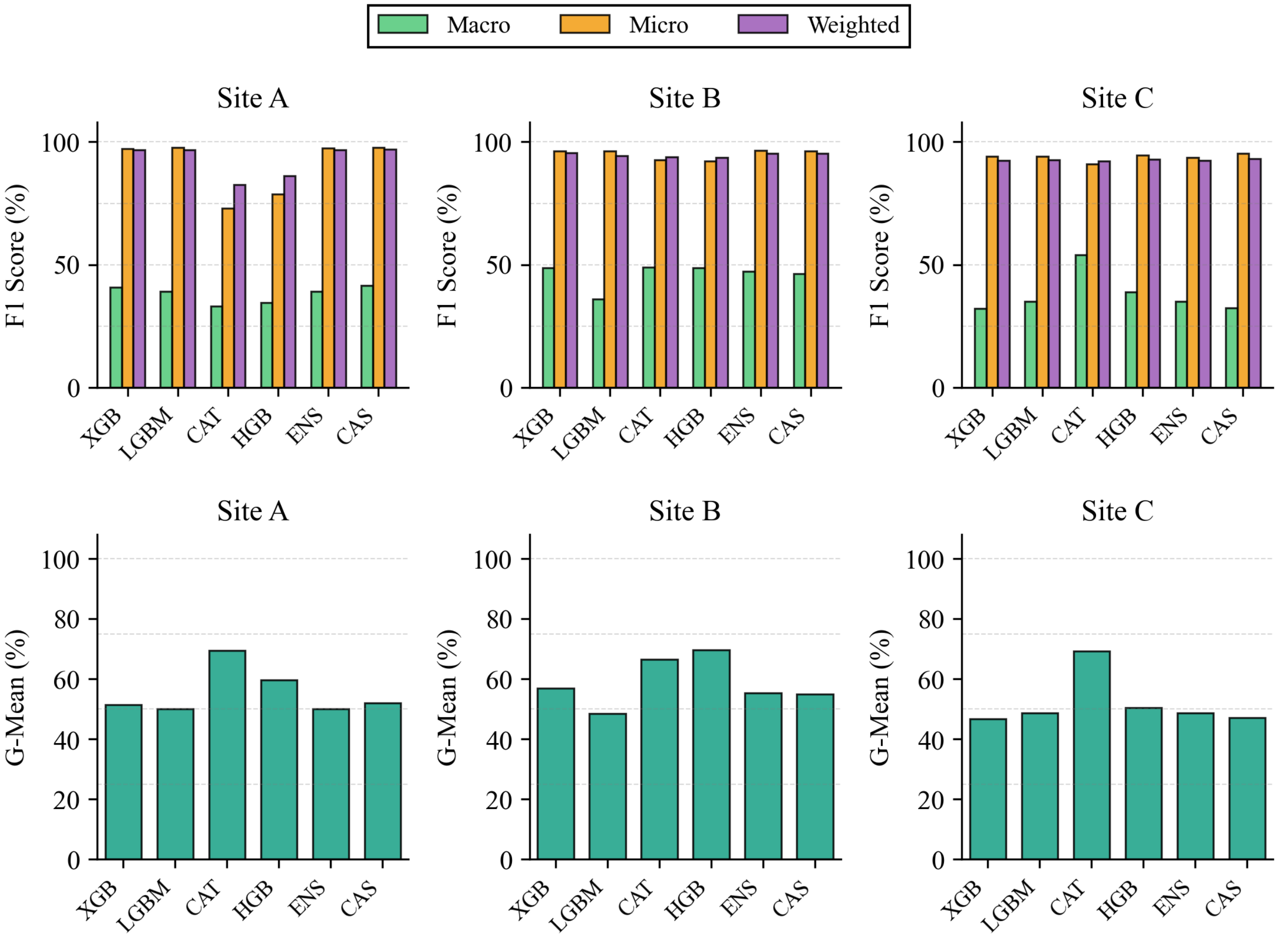

Fig. 2. Classification metrics across sites. (Top) F1-score breakdown showing macro, micro, and weighted averages. (Bottom) G-Mean scores. The cascade classifier achieves 50-52% G-Mean, best balancing performance across event types.

References

[1] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.

[2] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[3] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[5] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[6] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “Catboost: unbiased boosting with categorical features,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.